Pearls in Life

Technical reflections and records of thoughts

- RSS

Navigate...

- Home
- Projects
- Be Productive!
-
- Blog
- Archives

Simulate Random MAC Protocol in NS2 (Part IV)

Dec 15th, 2013

Now we have designed the simulator, add a new MAC protocol to NS2, and implement the Random Resend MAC protocol, the final part will be analyze the trace file to measure the performance of our new protocol.

Format of the Trance Entry

One line in the trace file may look like this.

```
1 s 0.010830867 _70_ MAC  --- 27 cbr 148 [0 46000000 8 0] ------- [70:0 0:0 32 0] [0] 0 0
2 |      |        |   |        |  |   |
3 |      |        |   |        |  |   +----- Packet Size
4 |      |        |   |        |  +--------- Traffic Type
5 |      |        |   |        +------------ Packet UID
6 |      |        |   +--------------------- Layer
7 |      |        +------------------------- Node ID
8 |      +---------------------------------- Time
9 +----------------------------------------- Event Type
```

You can find more details about the trace format here.

Trace Filtering

In this project, we're only interested the traces that

- From MAC layer, and
- With CBR traffic type
- With Event Type in "Send"(s), "Receive"® and "Drop"(D)

So we first filter those not-so-interesting traces out.

```
1  def filter(trace_file) :
2    with open(trace_file, 'r') as f :
3      raw_lines = f.read().split('\n')
```

```
4
5   print("%s raw traces found." % (len(raw_lines)))
6
7   traces = []
8   for line in raw_lines :
9     fields = line.split(' ')
10    if fields[0] not in ['s', 'r', 'D'] :
11      continue
12    if not (fields[3] == 'MAC') :
13      continue
14    if not (fields[7] == 'cbr') :
15      continue
16
17    traces.append({'action': fields[0], 'node': fields[2], 'pid': int(fields[6])})
18
19
20  print("%s filtered traces found." % (len(traces)))
21  return traces
```

Delivery Probability

To calculate the delivery probability, we need to know

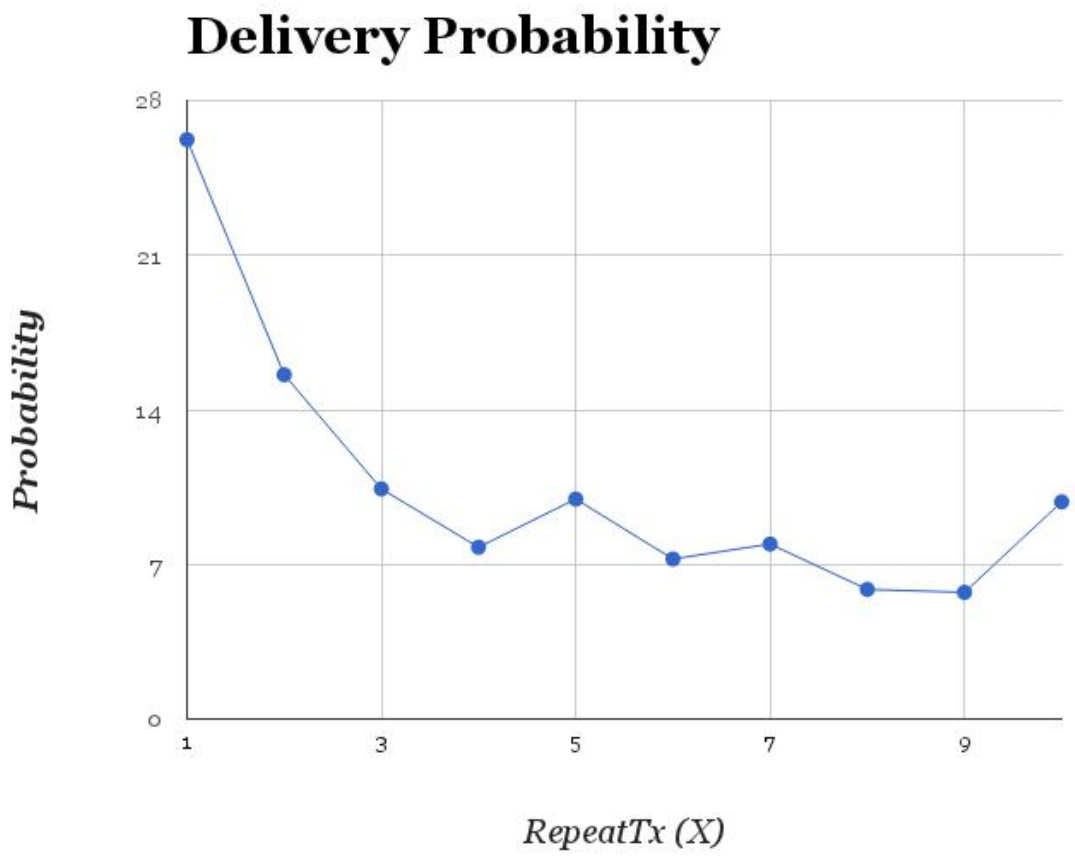
- How many unique packets are sent out by all source nodes?
- How many unique packets received by the sink node?

These two metrics can be easily obtained as following:

```
1 nodes = set(t['node'] for t in traces)
2 print("%s nodes found." % (len(nodes)))
3
4 sent = len(set(t['pid'] for t in traces))
5 recv = len(set(t['pid'] for t in traces if t['node'] == SINK_NODE and t['action'] == 'r'))
6
7 print("sent: %d, recv: %d, P: %.2f%%" % (sent, recv, float(recv)/sent*100))
```

Remember that we use `LossMonitor` as sink? Now is the time to cross-reference the results here with the ones from the stats file. The total received packets number should match.

The final delivery probability w.s.t the repeat count `X` is somehow like this in my case (packet size is 16 bytes).



Note that somehow this is not the ideal probability distribution. Please refer to this paper for theoretical analysis and also simulation results.

QoMOR: A QoS-aware MAC protocol using Optimal Retransmission for Wireless Intra-Vehicular Sensor Networks

---

Change History

- **Wed Dec 25 15:16:07 2013**
  add related paper
- **Sun Dec 15 16:02:08 2013**
  new post

View on Github

ns2, python, trace

Posted by Jinghao Shi Dec 15th, 2013 In Category: network Tagged with: mac,

Related Posts

- Simulate Random MAC Protocol in NS2 (Part I)
- Simulate Random MAC Protocol in NS2 (Part III)
- OS161 Swapping
- OS161 Tool Chain Setup
- Simulate Random MAC Protocol in NS2 (Part II)

« Simulate Random MAC Protocol in NS2 (Part III) OS161 Tool Chain Setup »

Comments

Did you know...

Chuck Norris protocol design method has no status, requests or responses, only commands.

Blog Stats

59 hits

Popular Posts

- OS161 TLB Miss and Page Fault
- OS161 Coremap
- OS161 fork System Call
- OS161 File System Calls
- OS161 execv System Call

Recent Posts

- Tap Notification to Send Email
- Sum of N Largest Numbers in Google Spreadsheet
- OS161 Tool Chain Setup
- Simulate Random MAC Protocol in NS2 (Part IV)
- Simulate Random MAC Protocol in NS2 (Part III)

Tags

AlertDialog C DownloadManager MLFQ SO_REUSEADDR addrspace backup beamer bind binutils bison bmake c c++ caption cat chrome close compare_xml coremap cron cscope ctags dispatch django dropbox dup2 email exit fd_set fdtable fedora figure function gcc gdb getifaddrs glic google calendar graphicspath graphicx grep heap intent jekyll lfs libpopller linux lock lseek mac migration notification ns2 open ota page fault page table pdflatex pid plugin popular posts prototype python query read readline rsync ruby rwlock sbrk scheduling scp sed select service signapk socket sort spreadsheet ssh stack swap synchronization sys161 syscall table tcl texinfo tlb toolchain undodir vimrc vm waitpid wordpress write yacc yum sty zip

Categories

Android (6)
errors (9)
latex (8)
linux (7)
network (8)
octopress (5)
os161 (25)
tricks (1)
vim (4)

GitHub Repos

- latex.template.slides

  Beamer template

- dotfiles

  Various configuration files

- jhshi.github.com

  Personal Blog

- latex.template.acm-proc

  Latex template for course project proposals, reports.

- latex.template.homework

  Latex template for homework.

@jhshi on GitHub

Stack Overflow

Page Link