برنامه آموزشی قابل درک GLOMOSIM

چکیده:

سند زیردرصد نشان دادن یک برنامه آموزشی ساده می باشد که برای استفاده و شبیه سازی شبکه های بی سیم در GLOMOSIM و همچنین ساختار اصلی شبیه ساز به کار می رود .ما تلاش می کنیم که به روش ثابتی اطلاعات را از چندین منبع جمع آوری نمائیم: مانند کاغذ های پیش نویس ، اطلاعات به روز فایل های README نرم افزار و ایمیل های موجود در فهرست ایمیل GLOMOSIM. نسخه نرم افزای که در این سند مورد بحث قرار گرفته است ۲/۰۳ برای سیستم عامل های یونیکس می باشد،اگر چه بسیاری از مفاهیم برای سیستم عامل های دیگر نیز کاربرد دارد.

فهرست مطالب:

۱. GLOMOSIM

۱-۱.استفاده از شبیه ساز GLOMOSIM

۱-۲ .ابزار برنامه نویسی

۲.تنظیم سناریو

۲-۱.فایل ساختار اصلی

۲-۲.فایل ساختار کاربردی

۲-۳.تنظیم دامنه انتقال

۲-۴.جابجا پذیری ساختار

۳.برخی نمونه های شبیه سازی

۳-۱.شبیه ساز پروتکل های MAC

۳-۱-۱.نمونه اول :مسئله پایانه مخفی

۳-۱-۲.نمونه دوم: مسئله پایانه بیان شده

۳-۲.تحلیل نتایج

۴.نکاتی درمورد پروتکل های مسیر یابی در GLOMOSIM

۴-۱.AODV

FSR.۲-۴

WRP.۳-۴

۵.راهنمای طرح نرم افزار GLOMOSIM

۵-۱.کد گذاری در GLOMOSIM

۵-۲.ساختار GLOMOSIM

۵-۳.طراحی GLOMOSIM

۵-۳-۱.زمان سنج

۵-۴.چگونگی اضافه کردن یک پروتکل جدید

۶.نصب و خطایابی

۶-۱.نصب

۶-۲.ابزار برنامه نویسی

.....

۱- GLOMOSIM

شبیه ساز سیستم اطلاعاتی سیار جهانی(GLOMOSIM) محیط شبیه ساز مقیاس پذیری برای شبکه های ارتباطی تلفن کابلی و بی سیم می باشد. GLOMOSIM از قابلیت شبیه سازی وقایع مجزا که توسط پارسک فراهم شده است ،استفاده می کند. GLOMOSIM شبکه ها را با صدها گره که با قابلیت ارتباطات ناهمگن به هم مربوط می شوند. شبیه سازی می کند که شامل چند بخشی ارتباطات نا متقارن با استفاده از انتشار امواج رادیویی ،ارتباطات بی سیمی چند بخشی با استفاده از شبکه بندی موقت و پروتکل های اینترنت سابق می باشد. جدول زیر مدل های GLOMOSIM را که اخیرا در هر لایه ئ اصلی قابل دسترس هستند را فهرست بند نموده است .

روش جمع آوری گره شبکه برای GLOMOSIM تعریف می شود تا فوائد قابل توجهی از عملکرد شبیه سازی را ارائه دهدمقدار دهی اولیه به هر گره شبکه به عنوان یک عضو جدا قابلیت مقیاس پذیری را محدود می کند زیرا نیازمندی های حافظه با افزایش تعداد گره های شبکه برای یک مدل به طور نمایشی افزایش می یابد. با جمع اوری گره شبکه عضو واحد می تواند چندین گره شبکه را در سیستم شبیه سازی کند روش جمع اوری گره شبکه نشان می دهد که تعداد گره ها در سیستم می تواند افزایش یابد در حالی که تعداد عضو ها در شبیه سازی حفظ می شوند هر عضو در

GLOMOSIM نشان دهنده ئ محیط جغرافیایی شبیه سازی می باشد بنابراین گره های شبکه که عضو خاصی را نشان می دهد با موقعیت فیزیکی گره ها تعیین می شوند .

۱-۱. استفاده از شبیه ساز GLOMOSIM

پس از نصب موفقیت امیز GLOMOSIM یک شبیه ساز می تواند با اجرای دستور زیر در زیر شاخه BIN اغاز می شود:

$$./\text{glomosim} \quad < inputfile >$$

(INPUTFILE) شامل پارامتر های ترکیب بندی برای شبیه سازی می باشد (یک نمونه از چنین فایلی CONFIG.IN می باشد) فایلی با نام GLOMO.STST در پایان شبیه سازی تولید می شود و شامل تمام امارهای تولید شده است .

۱-۲. ابزار برنامه نویسی:

GLOMOSIM دارای یک ابزار برنامه نویسی است که به خاطر کد گذاری در JAVA سخت افزاری مستقل می باشد .برای مقدار دهی به ابزار برنامه نویسی ،می بایست از کتاب راهنمای

JAVA-GUI مانند JAVA GLOMO MAIN استفاده کنیم از این ابزار می توان برای خطا زدایی و تغییر مدل هایاستفاده کرد مثل :توقف از سر گرفتن و اجرای مرحله ،انتقال سبک ها را نشان می دهد گره های سیار را در رنگ های متفاوت نشان دهید و آمار را نشان دهید.

لایه شبکه رادیویی در ابزار برنامه نویسی به صورت زیر نشان داده شده است :وقتی یک گره سبک را انتقال می دهد خط ارتباطی زرد از این گره به تمامی گره های موجود در دامنه کشیده می شود .هنگامی که هر گره سبک را دریافت می کند این خط ارتباطی پاک می شود و یک خط سبز برای دریافت موفق و یک خط قرمز برای دریافت ناموفق کشیده می شود هیچ تمایزی بین نوع سبک های متفاوت وجود ندارد .(سبک های کنترل ،در مقابل سبک های منظم و غیره ....)

۲- تنظیم یک سناریو

۲-۱.فایل ساختار اصلی

پارامترهای ساختار اصلی برای تنظیم سناریو در فایل config.in تعریف شده است این پارامتر ها به صورت زیر است:

مقادیر پیش فرض این پارامترهادر فایل config.in عبارتنداز:

پارامترNODE-PLACEMENT را می توان با مقادیر زیر تعیین کرد :RANDOM (گره ها به شکل تصادفی در زمین فیزیکی قرار داده شده اند)UNIFORM (براساس تعداد گره ها در شبیه سازی است و محیط فیزیکی به تعدادی از سلول ها تقسیم می شود در هر سلول یه گره به طور تصادفی قرار داده می شود).GRID (جایگاه گره در ۰و۰ شروع می شود و در شکل شبکه بندی با هر گره GRID-UNIT جدای از گره های مجاورش قرار داده می شود تعداد گره ها می بایست مربع یک عدد صحیح باشد)و FILE (موقعیت گره ها از NODE-PLACEMENT خواند می شود). مقادیر پیش فرض این پارامترها در فایل CONFIG .IN و یک نمونه از فایل NODES.INPUT عبارتنداز :

```
# NODE-PLACEMENT      FILE
# NODE-PLACEMENT-FILE ./nodes.input
# NODE-PLACEMENT      GRID
# GRID-UNIT           30
# NODE-PLACEMENT      RANDOM

  NODE-PLACEMENT      UNIFORM
```

```
# Example of the NODES.IN file
# Format: nodeAddr 0 (x, y, z)
# The second parameter is for consistency with the mobility trace format

0  0 (20.2,  0.9, 0.11)


1  0 (20.3, 30.8, 0.01)
2  0 (20.4, 60.7, 0.12)
```

اگر پارامتر MOBILITY برای NOTE تنظیم شود هیچ حرکتی از گره ها در مدل وجود نخواهد داشت .ساختارحرکت گرها دربخش[2-4] بیشتر توضیح داده شده است:

پارامتر PROPAGATION-PATHLOSS مدل اتلاف قدرت در مسیر خواص را مشخص می کند ،مدل های قابل دسترس در GLOMOSIM ،FREE-SPACE می باشد که دارای (توان از دست رفته ،سیگما) =(۰و۰و۲) می باشد .

مدل TWO-RA2 از فضای ازاد استفاده می کند و اتلاف قدرت در مسیر خواص (۰و۰و۲) برای دید نزدیک و برای زمین طراحی (۰و۰و۴)برای دید دور بکار می رود ارتفاع انتن در مدل مورد نظر به سختی کد گذاری می شود (۱/۵متر)مقادیر پارامتر radio-type عبارتند از :radio-accnotse که به مدل رادیوی استاندارد اشاره دارد در حالی که RADIO-NONOISE به مدل رادیویی انتزاعی اشاره

دارد (RADIO-NONOISEقابل قابل مقایسه با نسخه اخیر (۵.طو ۲/۱) مدل رادیویی NS-2 می باشد).
مقادیر پیش فرض این پارامتر ها در فایل CONFIG-IN عبارتند از:

```
   PROPAGATION-LIMIT        -111.0

#  PROPAGATION-PATHLOSS     FREE-SPACE
   PROPAGATION-PATHLOSS     TWO-RAY
#  PROPAGATION-PATHLOSS     PATHLOSS-MATRIX

   NOISE-FIGURE             10.0
   TEMPARATURE              290.0

   RADIO-TYPE              RADIO-ACCNOISE
#  RADIO-TYPE              RADIO-NONOISE

   RADIO-FREQUENCY         2.4e9
   RADIO-BANDWIDTH         2000000
```

در پارامتر RADIO –RX-TYPE ،هنگامی که پارامتر SNR-BOUNDED استفاده می شود و در صورتی که سیگنال NOISE RA (SNR) بیشتر از RADIO-RX-SNR-THRESHOLD (درDB) باشد سیگنال بدونه خطا دریافت می شود در غیر این صورت بسته اطلاعاتی کاهش می یابد -RADIO-RX SNR-THRESHOLD را می یابست مشخص کرد . پارامتر BERBASED میزان خطای بایت را در جدول SNR-BER جستجو کرده و با فایل BER-BPSK.IN به صورت زیر نوشته می شود :

```
  RADIO-RX-TYPE            SNR-BOUNDED
  RADIO-RX-SNR-THRESHOLD   10.0
# RADIO-RX-SNR-THRESHOLD   8.49583

# RADIO-RX-TYPE            BER-BASED
# BER-TABLE-FILE           ./ber_bpsk.in

  RADIO-TX-POWER           15.0
  RADIO-ANTENNA-GAIN        0.0
  RADIO-RX-SENSITIVITY    -91.0
  RADIO-RX-THRESHOLD      -81.0
```

```
# Example of BER_BPSK.IN file:

0.000000        5.000000e-01
0.020000        4.207403e-01
0.040000        3.886487e-01
0.060000        3.645172e-01
```

مقادیر پیش فرض این پارامترها در فایل CONFIG.INبه صورت زیر است :

```
  MAC-PROTOCOL          802.11
# MAC-PROTOCOL          CSMA
# MAC-PROTOCOL          MACA

# MAC-PROTOCOL          TSMA
# TSMA-MAX-NODE-DEGREE       8

# MAC-PROPAGATION-DELAY 1000NS
# PROMISCUOUS-MODE      NO
```

```
NETWORK-PROTOCOL    IP
NETWORK-OUTPUT-QUEUE-SIZE-PER-PRIORITY 100

    # RED-MIN-QUEUE-THRESHOLD 150
    # RED-MAX-QUEUE-THRESHOLD 200
    # RED-MAX-MARKING-PROBABILITY 0.1
    # RED-QUEUE-WEIGHT .0001
    # RED-TYPICAL-PACKET-TRANSMISSION-TIME 64000NS

      ROUTING-PROTOCOL    BELLMANFORD
    # ROUTING-PROTOCOL    AODV
    # ROUTING-PROTOCOL    DSR
    # ROUTING-PROTOCOL    LAR1
    # ROUTING-PROTOCOL    WRP
    # ROUTING-PROTOCOL    FISHEYE
    # ROUTING-PROTOCOL    ZRP

    # ZONE-RADIUS         2

    # ROUTING-PROTOCOL    STATIC
    # STATIC-ROUTE-FILE   ROUTES.IN
      APP-CONFIG-FILE   ./app.conf
```

در نهایت ،پارامتر های زیر تعیین می کنند که ایا توجه ما بیشتر بر روی امار و ارقام سیگنال یا سطح چند گانه است یا خیر پارامتر هایی که با 2ES مشخص شده اند یعنی شبیه سازی اماری را برای لایه ویژه فراهم نموده است :

```
APPLICATION-STATISTICS          YES
TCP-STATISTICS                  NO
UDP-STATISTICS                  NO
```

```
ROUTING-STATISTICS              NO
NETWORK-LAYER-STATISTICS        NO
MAC-LAYER-STATISTICS            NO
RADIO-LAYER-STATISTICS          NO
CHANNEL-LAYER-STATISTICS        NO
MOBILITY-STATISTICS             NO

# GUI-OPTION: YES allows GloMoSim to communicate with the Java Gui Visual Tool

GUI-OPTION   YES
GUI-RADIO    YES
GUI-ROUTING  YES
```

۲-۲.کاربرد فایل ترکیب بندی

کاربد هایی مثل FTP وTELENT در این فایل ترکیب بندی می شود. مولد های ترافیکی (تولید کننده ازدحام اطلاعات)که اخیرا در دسترس قرار گرفته اند شامل FTP، HTTP،CBR،TALENT،FTP/GENERIC می باشند FTP برای شبیه سازی پروتکل انتقال فایل از TCPLIB استفاده می کند . به منظور استفاده از FTP قالب بندی زیر مورد نیاز است

FTP<SRC>DEEST><ITEMS TO SEND><START TIME>

که در ان SRC یک گره کلاینت یا مشتری است <DEST> گره سرور می باشد ،<ITEM TO SEND> برای این است که چه تعداد قلم های اطلاعاتی کاربردی فرستاده می شود و <START TIME> زمان شروع FTP در طول شبیه سازی می باشد . در صورتی که <ITEM TO SEND> روی عدد صفر تنظیم شود ، FTP از TCPLIB استفاده می کند تا میزان قلم های اطلاعاتی کاربردی را برای فرستادن تعیین کند اندازه هر قلم همیشه به طور تصادفی و توسط TCPLIB تعیین می شود توجه داشته باشید که اصطلاح قلم در سطح کاربردی معادل است با اصطلاح سبک در سطح شبکه و بسته ی داده ای در سطح MAC برخی نمونه ها عبارتند از :

- FTP 01 10 OS :در اغاز شبیه سازی ۱۰ قلم که اندازه هر یک از انها به شکل تصادفی توسط TCPLIB تعیین می شود گره صفر را به گره یک ارسال می کند
- FTP 010100S پس از گذشت ۱۰۰ ثانیه از شبیه سازی تعداد قلم ها توسط TCPLIB انتخاب می شودو گره صفر را به گره ۱ ارسال می کند .

FTP/GENERIC برای انتقال فایل شبه سازی شده از TCPLIB استفاده نمی کند اما در عوض کلاینت یا سرویس گیرنده قلم های اطلاعاتی را به سرور ارسال می کند بدونه انکه سرور اطلاعات کنترل را در پشت سرویس گیرنده بفرستد به منظور استفاده از FTP/GENERIC قالب بندی زیر مورد نیاز می باشد

FTP/GENERIC<SRC><DEST><ITEM TO SEND><ITEM SIZE><START TIME>

که در ان <SRC> گره سرویس گیرنده <DEST> گره سرور<ITEM TO SEND> تعداد قلم های اطلاعاتی سطح کاربر برای ارسال <ITEM SIZE> اندازه هرقلم سطح کاربری می باشد و <START TIME> زمان شروع FTP/GENERIC در طول شبیه سازی است و <END TIME> زمان پایان FTP/GENERIC می باشد .در صورتی که <ITEM TO SEND> روی صفر تنظیم شود FTP/GENERIC تا مشخص شدن <END TIME> یا تا پایان شبیه سازی اجرا خواهد شد . در صورتی که <ITEM TO SEND> و <END TIME> هر دو بزرگتر از صفر باشند FTP/GENERIC تا انجام<ITEM TO SEND> و رسیدنبه <END TIME> یا پایان شبیه سازی اجرا خواهند شد برخی از این نمونه ها در زیر آورده شده است :

- FTP/GENERIC 011014600S600S :گره صفر ، گره ۱ و ۱۰قلم اطلاعاتی ۱۴۶۰بایتی را در شروع شبیه سازی تا ۶۰۰ ثانیه پس از شبیه سازی ارسال می کند در صورتی که ۱۰ قلم قبل از ۶۰۰ ثانیه ارسال شوند قلم های دیگر فرستاده نمی شوند
- FTP/GENERIC011014600S0S : گره صفر ،گره یک و ۱۰ قلم اطلاعاتی ۱۴۶۰ بایتی را در شروع شبیه سازی و تا پایان ان ارسال می کند اگر ۱۰ قلم تا پایان شبیه سازی ارسال شود قلم های دیگر ارسال نمی شوند
- FTP/GENERIC 01014600S0S : گره صفر به طور مداوم قلم های ۱۴۶۰ بایتی گره یک را در شروع شبیه سازی تا پایان ان ارسال می کند.

TELNET برای شبیه سازی پروتکل تل نت از TCPBIL استفاده می کند به منظور استفاده از TELNET قالب بندی زیر مورد نیاز است:

TELNET<SRC><DEST><SESSI55>

که در ان <SRC> گره سرویس گیرنده <DEST> گره سرور <SESSION DERATION> مدت دوره کارایی تل نت <START TIME> زمان شروع تل نت در طول شبیه سازی است در صورتی که <SESSION DERATION> روی عد صفر تنظیم شود TELNET از TCPLIB استفاده می کند تا به شکل تصادفی مدت دوره ی کاری تل نت را تعیین کند فاصله بین قلم های تل نت با TCPLIB تعیین می شوند برخی این نمونه ها عبارتند از :

- TELNET 01100S0S : گره صفر ترافیک یا ازدحام تل نت گره یک را برای دوره ی ۱۰۰ ثانیه در شروع شبیه سازی تعیین می شود ارسال می کند .
- TELNET 0110S0S : گره صفر ترافیک تل نت گره یک را برای دوره ای که به طور تصادفی توسط TCPLIB در شروع شبیه سازی تعیین می شود ،ارسال می کند .

CBR مولدی با بایت ثابت را شبیه سازی می کند به منظور استفاده از CBR قالب بندی زیر مورد نیاز است:

```
CBR <src> <dest> <items to send> <item size> <interval> <start time> <end time>
```

که در این قالب بندی <SRC> گره سرویس گیرنده <DEST> گره سرور <ITEM TO SEND> تعداد قلم های لایه کاربردی برای ارسال <ITEM SIZE> اندازه هر قلم لایه کاربری<INTERRAL> زمان حرکت بین قلم های اطلاعاتی لایه ی کاربردی <START TIME> زمان شروع CBR در طول شبیه سازی <END TIME> زمان پایان CBR در طول شبیه سازی می باشد در صورتی که <ITE TO SEND> بر روی عدد صفر تنظیم شود CBR تا انتقال <ITEMS TO SEND>یا تا پایان شبیه سازی اجرا می شود اگر <ITEM TO SEND> و<END TIME> هر دو بزرگتر از صفر باشند CBRتا انجام دستور <ITEMS TO SEND> و رسیدن<END TIME> یا پایان شبیه سازی اجرا خواهد شد برخی از نمونه ها به صورت زیر می باشند :

- CBR 01 10 1460 1S 0S0S600S : گره صفر ۱۰ قلم اطلاعاتی ۱۴۶۰بایتی گره یک را در شروع شبیه سازی تا سقف ۶۰۰ ثانیه برای عمل شبیه سازی ارسال می کند زمان حرکت درونی برای هر قلم یک ثانیه می باشد اگر ۱۰ قلم اطلاعاتی قبل از گذشتن ۶۰۰ ثانیه فرستاده شود قلم های دیگر ارسال نمی شوند .
- CBR 010114601S0S0S :گره صفر به طور مداوم قلم های ۱۴۶۰ بایتی گره یک را در شروع شبیه سازی تا ۶۰۰ ثانیه ارسال می کن زمان حرکت درونی برای هر قلم یک ثانیه است.
- CBR 01014601S0S0S :گره صفر به طور مداوم قلم های ۱۴۶۰بایتی گره یک را در شروع شبیه سازی تا پایان شبیه سازی ارسال می کند ، زمان حرکت درونی برای هر قلم یک ثانیه است .

HTTP سرور های شبکه ارتباطی TCP و کلاینت را شبیه سازی می کند قالب بندی زیر کاربرد ان را برای سرورها توصیف می کند : HTTP<ADDRESS>

که در اینجا <ADDRESS> ادرس گره ای است که صفحات وب را پوشش می دهد برا کلاینت های HTTP قالب بندی زیر استفاده می شود :

```
HTTP <address> <num_of_server> <server_1> ... <server_n> <start> <thresh>
```

در اینجا <ADDRESS> ادرس گره ای است که در ان کلاینت قرار دارد <NUM OF SERVER> تعداد نشانی های سرور که به صورت <SERVER I>، <SERVER N> می باشد که نشانی های گره سرور هایی است که این کلاینت بین صفحات پاسخ انتخاب خواهد شد برای هر ادرس یا نشانی می بایست خطوط<ADDRESS> وجود داشته باشد.

<start> زمان شروع برای وقتی است که کلاینت صفحات پاسخ را اغازمی کند<thresh> سقفی است که برروی مقدار زمان تفکر برای یک کلاینت مشخص شده است .اثر شبکه بندی براساس مقدار پیمانه زمان است و این استانه برای تعیین زمان تفکر استفاده می شود یک نمونه از این مطلب در زیر گفته شده :

HTTPD 2
HTTPD 5
HTTPD 8
HTTPD 11
HTTP 1 3 2 5 11 10S 120S

سرورهای HTTP روی گره های ۲-۵-۸و۱۱ وجود دارد یک کلاینت HTTP روی گره یک وجود دارد این کلاینت بین سرورهای [۲و۵و۱۱] انتخاب می شود و پس از ۱۰ ثانیه از گذشت زمان شبیه سازی این کلاینت شروع به جستجو می کند و به مدت دو دقیقه بعد از زمان شبیه سازی تفکر می کند .

۲-۳.تنظیم دامنه انتقال :

به علت پیچیدگی برخی روش ها انتقال های رادیویی توسط محیط اطراف تحت تاثیر قرار گرفته اند به همین خاطر پیش بینی اجرای سیستم و تعریف دامنه انتقال رادیویی یک گره مشکل می باشد دامنه رادیویی حداکثر فاصله میانگین در شرایط عملیاتی معمول بین دو گره می باشد فرایند عملیاتی معمول واستانداردی وجود ندارد که دامنه را اندازه گیری کند (به جز در فضای ازاد که بی فایده هم می باشد) بنابراین

نمی توانیم تولیدات متفاوت را از دامنه هایی که در دستگاه های سیار نشان داده شده اند مقایسه کنیم

اگر ما بخواهیم گره های سیار را بر حسب عملکرد دامنه مقایسه کنیم ما می بایست با دقت به مقادیر حساسیت و سیره انتقال یافته نگاه بیاندازیم . چندین ویژگی قابل اندازه گیری در مورد سخت افزار وجود دارد که عملکرد تولیدات را در ان زمینه نشان می دهد نیروی انتقال یافته قدرت انتشار را اندازه گیری شده در وات می باشد (یا میلی وات) قوانین دولتی این قدرت را محدود می کند اما دارای یک قدرت انتقالی بالاست که باتری ها را سریعتر تخلیه می کند با این وجود داشتن قدرت انتقالی بالا به حذف سیگنال های قوی تر در باند کمک می کند حساسیت اندازه گیری ضعیف ترین

سیگنالی است که ممکن است به طور قابل اعتمادی برای کانالی که توسط دریافت کننده ایجاد شده است شنیده می شود(همچنین قادر است که بیت ها را از انتن بخواند البته با احتمال خطای خیلی پایین) این عملکرد دریافت کننده را نشان می دهد که هرچقدر مقدار ان پایین تر باشد سخت افزار بهتری است مقادیر معمولا حدود 80dbm- است (برای مثال 90dbm – بهترین نمونه می باشد)روش احتمالی برای تعیین دامنه رادیویی انتقال در GLOMOSIM به صورت زیر است :

۱-مدل اتلاف قدرت در مسیر خاص انتشار را تنظیم کنید (پارامتر PROPAGATION-PATHLOOS)

۲-قدرت دریافت انتن مقصد را ثابت نگه دارید (پارامتر RADIO-RX-THRESHOLD)

۳-فاصله را ثابت نگه دارید و قدرت انتقال یافته را مطابق با مدل انتشار محاسبه کنید

۴-این مقدار را با پارامتر RADIO-TX-POWER تنظیم کنید

GLOMOSIM مدل انتشار های زیر رادارد مدل فضای ازاد و مدل انعکاس زمین (یا دو شعاع) مدل انتشاری فضای ازاد برای پیش بینی قدرت سیگنال دریافتی وقتی استفاده می شود که انتقال دهند و دریافت کننده خط دید واضح وبدون مانعی داشته باشند این مدل پیش بینی می کند که قدرت انتقال در تناسب با مربع فاصله کم تر می شود بر طبق این مدل معدله فضای ازاد برای انتن های غیر مشابه به صورت زیر می باشد:

$$P_r = P_t(\frac{\lambda}{4\pi d})^n G_t G_r$$

که در ان Pr قدرت دریافت شده Pt قدرت انتقال یافته Y طول موج حامل d فاصله بین انتقال دهند و دریافت کننده د ضریب اتلاف قدرت در مسیر خاص Gt انتن بدست امده در دریافت کننده است .برای انت ایزو تروپیک ایده ال معادله ی به شکل زیر است :

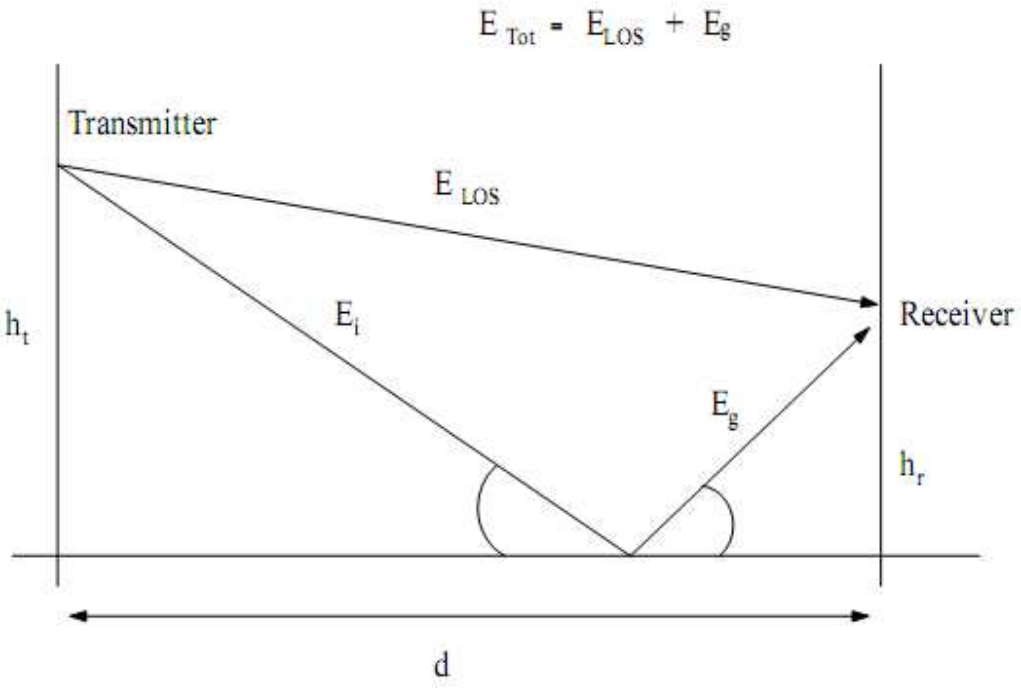
$$\frac{P_t}{P_r} = \frac{(4\pi d)^2}{\lambda^2} = \frac{(4\pi f d)^2}{c^2}$$

که C سرعت نور و F فرکانس می باشد

مدل انعکاس زمین (دو شعاع) هر دو روش مستقیم و یک روش انتشار انعکاس یافته زمین را بین انتقال دهنده ودریافت کننده مشخص می کند این مدل پیش بینی می کند که قدرت دریافتی با فاصله بالا رفته پایین می رود این مدل سرعت خیلی بالایی نسبت به فضای ازاد تجربه شده دارد :

$$P_r = P_t \frac{h_t^2 h_r^2}{d^4} G_t G_r$$

کهh_r, h_t ارتفاع انتقال دهنده و انتن های دریافت کننده می باشند ودر glomosim این مقدار تا ۱/۵ متر کد گذاری می شود این مدل در شکل ۱ نشان داده شده است ضریب انتن (که در glomosim با پارامتر RADIO-ANTENNAGAIN نشان داده می شود) اندازه گیری جهت یک ا نتن می باشد ضریب انتن به عنوان اطلاعات خروجی توان تعریف می شود که در جهتی خاص با تولید هر جهت . باانت همه سویه مقایسه می شود برای مثال اگر یک انتن دارای ضریب 3db باشد انتن در جهت 3db یا ضریب ۲ می تواند انتن ایزوتروپیک را بهبود بخشیدتوان افزایش یافته در جهت داده شده منتشر می شود که برامدی از جهت های دیگر نیز می باشد .



در هر دو روش ما بایست کمبود یا تضعیف قدرت سیگنال را اندازه گیری کنیم دسی بل واحد اندازه گیری نسبت بین دو سطح سیگنال می باشد ضیب دسی بل با معادله زیر ارائه شده است :

$$G_{dB} = 10 \quad log_{10} \frac{P_{out}}{P_{in}}$$

به خاطر استفاده از اصطلاحات ضریب و کمبود (LOSS,GAIN) ناهماهنگی هایی در متن وجود دارد اگر مقدار GDB مثبت باشد ضریب واقعی توان ارائه می شود برای مثال ضریب 3db بدان معناست که توان دو برابر شده است در صورتی که مقدار GDB منفی باشد کمبود واقعی توان ارائه می شود برای مثال ضریب 3db- بدان معناست که توان نصف می شود با این وجود برخی از متن می گویند که این مقدارکمبودی از 3db- می باشد می تان ثابت کرد که ضریب منفی مطابق با اتلاف مثبت است مقادیر دسی بل به مقادیر نسبی یا تغییر در مقدار اشاره دارد نه به سطح مطلق معمول است که به سطح مطلق توان در دسی بل ها طوری اشاره شود که ضریب ها و اتلاف ها با اشاره به سطح سگنال اغاز محاسبه شوند dbw (دسی بل وات) به طور گسترده ای در کاربرد های زیر موج استفاده می شود مقدار یک وات به عنوا منبع انتخاب می شود و به شکل odbw تعریف می شود واحد رایج دیگر dbm است که از 1mw به عنوان مرجع استفاده می کند بنابراین odbm=1mw می باشد که در فرمول زیر ارائه شده است .

$$Power_{dbW} = 10 \quad log\frac{Power_W}{1W} \qquad Power_W = 10^{\frac{Power_{dBW}}{10}}$$

$$Power_{dbm} = 10 \quad log\frac{Power_{mW}}{1mW} \qquad Power_{mW} = 10^{\frac{Power_{dBm}}{10}}$$

برخی از نمونه های محاسبه نظری فاصله رادیویی مطابق با فضای ازاد و مدل های دو شعاعی به صورت زیر می باشند

```
I) Example 1.
            PROPAGATION-PATHLOSS =  FREE-SPACE
            PROPAGATION-LIMIT    =  -111 (dBm)
            RADIO-FREQUENCY      =   2.4 e 9 (hertz)
            RADIO-TX-POWER     =    15 (dBm)
            RADIO-RX-THRESHOLD   =   -81 (dBm)
            RADIO-ANTENNA-GAIN   =   0.0 (dBm)
```

بنابراین از اتلاف فضای ازاد انتن ایزوتروپیک استفاده می کنیم (radio-antenna-gain=0,0) که انتن ایزوتروپیک را نشان می دهد)Pt=15dbmb,Pr=-81dbm را به mw تبدیل می کنیم و فاصله را بدست می اوریم

$$Power_{mW} = 10^{\frac{Power_{dBW}}{10}} = 10^{1.5} = 31.622776mW = P_t$$

$$Power_{mW} = 10^{\frac{Power_{dBW}}{10}} = 10^{-8.1} = 7x10^{-9}mW = P_r$$

$$d = \frac{\sqrt{\frac{P_t}{P_r}c^2}}{4\pi f} = d = \frac{\sqrt{\frac{31.622776}{7x10^{-9}}(3x10^8)^2}}{4\pi(2.4x10^9)} = 668.57m$$

II) Example 2.

```
PROPAGATION-PATHLOSS =   TWO-RAY
PROPAGATION-LIMIT    =  -111 (dBm)
RADIO-FREQUENCY      =   2.4 e 9 (hertz)
RADIO-TX-POWER       =    15 (dBm)
RADIO-RX-THRESHOLD   =   -81 (dBm)
RADIO-ANTENNA-GAIN   =   0.0 (dBm)
```

مدل اتلاف قدرت در مسیر خاص به ما نشان می دهد که از فرمول مدل دو شعاعی استفاده کنم
Gt=Gr=odbm=1 بنابراین :

$$P_r = P_t \frac{h_t^2 h_r^2}{d^4} G_t G_r$$

$$d = \sqrt[4]{\frac{P_t h_t^2 h_r^2}{P_r}} = \sqrt[4]{\frac{(31.6227mW)(1.5)^2(1.5)^2}{7x10^{-9}}} = 388m$$

III) Example 3.

```
PROPAGATION-PATHLOSS =   TWO-RAY
PROPAGATION-LIMIT    =  -111 (dBm)
RADIO-FREQUENCY      =   2.4 e 9 (hertz)
RADIO-TX-POWER       =    15 (dBm)
RADIO-RX-THRESHOLD   =   -81 (dBm)
RADIO-ANTENNA-GAIN   =   10.0 (dBm)
```

این مربوط به مسئله قبلی است اما با پارامتر 10/0dbmradiop-antenna=gain می باشد ما ان را به مقدار قابل مقایسه با انت ایزوتوپریک تغییر می دهیم

$$
\begin{aligned}
G_{mW} &= 10^{\frac{G_{dBm}}{10}} = 10mW = 10 \\
d &= \sqrt[4]{\frac{P_t G_t G_r h_t^2 h_r^2}{P_r}} = \sqrt[4]{\frac{(31.6227mW)(10)(10)(1.5)^2(1.5)^2}{7x10^{-9}}} = 1229.749m
\end{aligned}
$$

در فهرست راهنمای BIN برنامه دامنه رادیویی دامنه رادیویی فایل ترکیب بندی را به دست می دهد برای مثال برای پیش نویس فایلCONFIG-IN داریم :

```
$>./radio_range config.in

radio range: 376.782m
Execution time :   0.0281 sec
Number of messages processed : 0
Number of context switches occurred : 6
Number of Local NULL messages sent : 0
Number of Remote NULL messages sent : 0
Total Number of NULL messages sent : 0
```

اجرای برنامه با پارامترهای ترکی بندی رادیویی متفاوت صورت می گیرد و نتایج زیر را بدست می اورد اگر چه ما میتوانیم دامنه رادیویی را با تعیین توان انتقال دهنده (RADIO-TX-POWER) یا توان دریافت کننده (RADIO-RX-THRESOHLD) تغییر دهیم اما تغییر توان انتقال دهنده تا حدی قابل توصیه می باشد زیرا توان دریافت کننده بستگی به محیط دریافت کننده بستگی به محیط رادیویی دارد در حالی که می توانیم توان انتقال دهنده را کنترل کنیم .

Radio Parameters					
PROPAGATION-LIMIT (dBm)	-111	-111	-111	-111	
PROPAGATION-PATHLOSS	Free-space	Two-Ray	Two-Ray	Two-Ray	
RADIO-FREQUENCY (hz)	2.4e9	2.4e9	2.4e9	2.4e9	
RADIO-TX-POWER (dBm)	15	15	15	15	
RADIO-ANTENNA-GAIN (dBm)	0	0	10	1	
RADIO-RX-SENSITIVITY (dBm)	-91	-91	-91	-91	
RADIO-RX-THRESHOLD (dBm)	-81	-81	-81	-81	
Radio Range (m)	627.625	376.782	1191.422	422.757	
Radio Parameters					
PROPAGATION-LIMIT (dBm)	-111	-111	-111	-111	-111
PROPAGATION-PATHLOSS	Two-Ray	Two-Ray	Two-Ray	Two-Ray	Two-Ray
RADIO-FREQUENCY (hz)	2.4e9	2.4e9	2.4e9	2.4e9	2.4e9
RADIO-TX-POWER (dBm)	15	1	-10	15	15
RADIO-ANTENNA-GAIN (dBm)	0	0	0	0	0
RADIO-RX-SENSITIVITY (dBm)	-61	-91	-91	-91	-91
RADIO-RX-THRESHOLD (dBm)	-81	-81	-81	-61	-101
Radio Range (m)	376.782	125.227	35.293	62.762	1191.492

۲-۴.ترکیب بندی سیار :

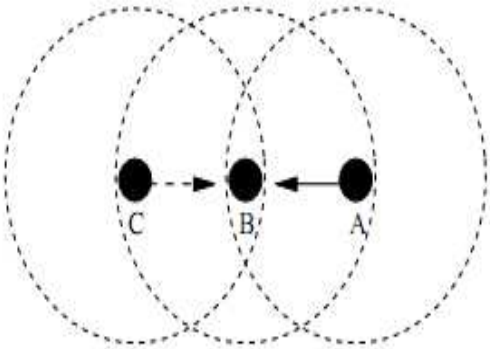
تنها مدل سیار قابل دسترس در GLOMOSIM مدل سیار WAYPOINT می باشد در این مدل یک گره به طور تصادفی مقصد را از زمین فیزیکی انتخاب می کند و در جهت ان مقصد با سرعتی یکنواخت بین پارامترهای SPEED-MIN-WP-MOBILITY-SPEED-MAY-WP-MOBILITY حرکت می کند پس از رسیدن به مقصد گره برای دوره ی زمانی PAUSE-WP-MOBILITY باقی می ماند .

اگر بخواهیم از الگوی سیار به جای قصحئ استفاده کنیم انگاه می بایست پارامتر MOBILITY-TRACE را به منظور نشان دادن GLOMOSIM مشخص کرد که حرکت های فردی را برای گره ها از فایل MOBILITY-TRACE-FILE می گیرد پارامتر MOBILITY-INTERNAL برای نشان دادن گره ها و به روز اوری موقعیت انها در هر دوره زمانی MOBILITY-INTERVAL استفاده می شود در حالی که MOBILYTY –D-UPDATE زمانی استفاده می شو که گره مو قعیت خودش را براساس فاصله به روز کند فلسفه GLOMOSIM تا حدودی با NS-2 متفاوت می باشد در NS-2 نقطه مقصد و سرعت ثابت را نشان می دهیم و دز GLOMOSIM نقطه مقصد و زمان رسیدن گره را مشخص می کنیم این ابزار سرعت ثابت را برای انجام این کار تعیین می کند پارامتر های سیار در فایل CONFIG.IN تعریف شده اند که به صورت زیر :

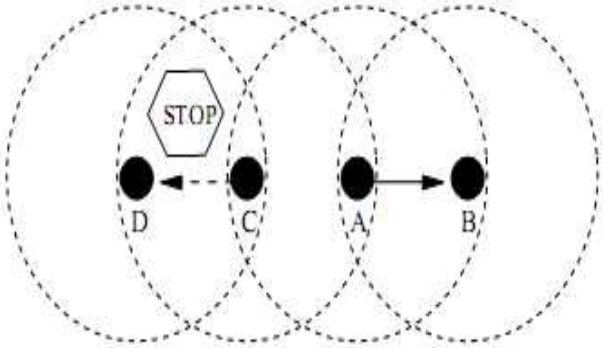
```
MOBILITY  NONE

# MOBILITY RANDOM-WAYPOINT
# MOBILITY-WP-PAUSE        30S
# MOBILITY-WP-MIN-SPEED    0
# MOBILITY-WP-MAX-SPEED    10

# MOBILITY TRACE
```



THE HIDDEN TERMINAL PROBLEM



THE EXPOSED TERMINAL PROBLEM

Figure 2: The Hidden and Exposed Terminal Problems

```
# MOBILITY-TRACE-FILE ./mobility.in

# MOBILITY PATHLOSS-MATRIX

# The following parameter is necessary for
# all  mobility models

MOBILITY-POSITION-GRANULARITY 0.5

# Example of MOBILITY.IN file
# mobility trace format: node-address simclock destination(x y z)
# All lines for a node must be sorted in time increasing order.

10 100S (200.0, 150.0, 0.2)
10 200S (200.0, 150.0, 0.2)
10 500S (250.0, 250.0, 0.2)
```

۳-برخی از نمونه های شبیه سازی

۳-۱.شبیه سازی پروتکل های MAC

GLOMOSIM دارای چندین پروتکل NAC است که در یک شبیه سازی مثل CSMA,MACA و ۸۰۲/۱۱ استفاده می شود در پروتکل CSMA(دستیابی چند گانه با حس حامل) یک گره درصدد انتقال می باشد و به واسطه اطلاعاتی گوش می دهد تا تعیین کند که ایا انتقال دیگری در جریان هست یا نه (حس حامل) در صورتی که واسطه اطللاعاتی انتقال مورد استفاده قرار گیرد گره در حال انتظار می ماند در غیر این صورت ممکن است منتقل شود متاسفانه CSMA با دو مکانیزم مجع محدود می شود مسائل نمایان و مخفی پایانه مسئله پایانه مخفی به این علت رخ می دهد که شبکه رادیویی ای که در معرض شبکه های دیگر مانند LAN قرار می گیرد از درجه بالای ضمانتی اتصال برخوردار نیست

بنابراین دو گره که اتصال به سومین گره را حفظ می کنند لزوما نمی توانند یک دیگر بپذیرند در شکل دو گره A در ارتباط با گره L3 می باشد که A در این اواخر در حال انتقال می باشد گره C درصدد ارتباط با گرهB می باشد در ادامه پروتکل CSMA گره C به واسطه اطلاعاتی گوش می دهد اما چونC انتقال گره A را کشف نمی کند فرمان نمایش ازاد واسطه اطلاعاتی را روش می کند در نتیجه C به واسطه اطلاعاتی دسترس می یابد که باعث تلاقی در گره B می شود .

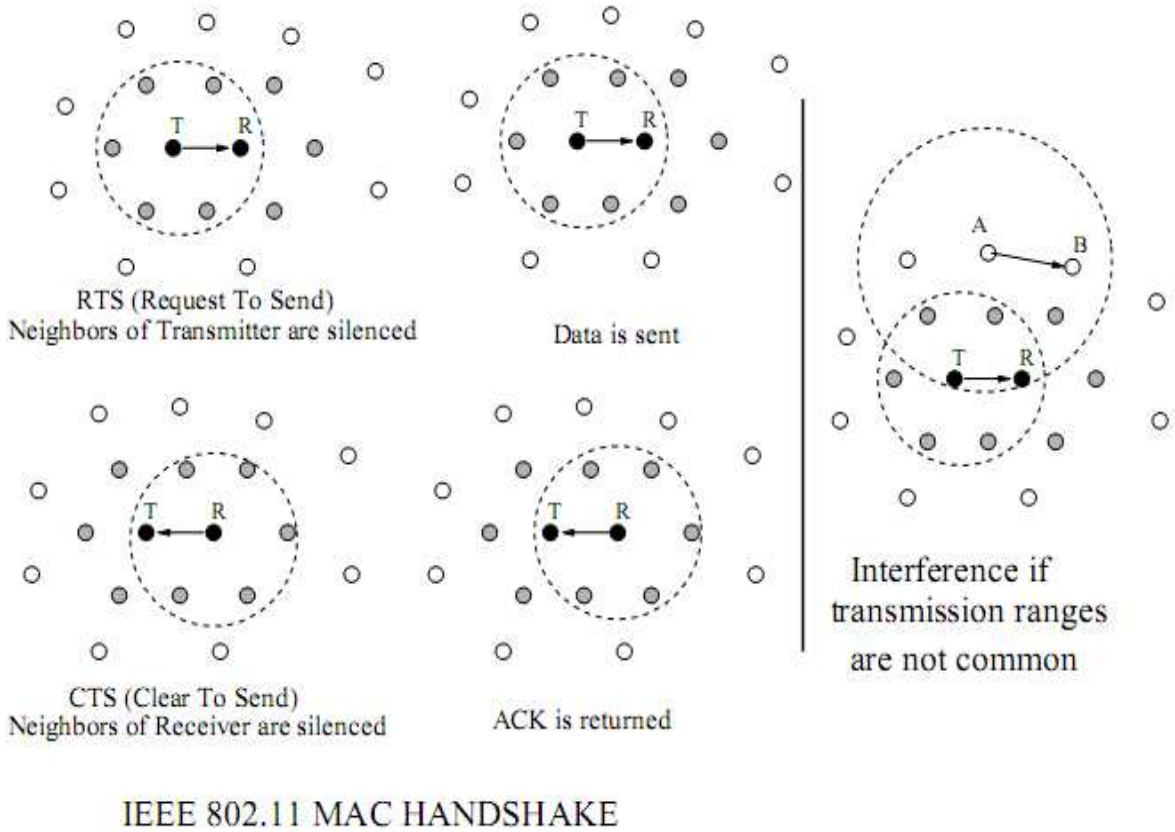


Figure 3: The RTS/CTS dialogue

دومین مسئله که در شکل ۲ نشان داده شده است پایانه نمایش زمانی رخ می دهد که به عنوان مثال گره A به گره B انتقال می یابد در حالی گره C می خواهد گره D را انتقال دهد در پروتکل CSMA گره C به واسطه اطلاعاتی گوش می دهد که گره A از دسترس به واسطه اطلاعاتی منتقل شده و به تعویق می افتد با این وجود دلیلی وجود ندارد که چرا گره C نمی تواند به طور همزمان با انتقال گره A انتقال یابند مانند انتقال گره C که با دریافت در گره B با توجه به فاصله بین دو گره تداخل نمی یابد . نکته ای که در اینجا وجود دارد این است که تلاقی در دریافت کننده رخ می دهد در حالی که پروتکل CSMA وضعیت واسطه اطلاعاتی را در انتقال دهنده برسی می کند به طور کلی مسئله پایانه مخفی قابلیت شبکه را با توجه به افزایش تعداد تلاقی کاهش می دهد در حالی که مسئله پایان نمایان قابلیت شبکه را با توجه به

تعویق غیر ضروری گره ها از انتقال کاهش می دهد تلاش هایی در جهت کاهش تاثیر این مسئله صورت گرفته است ضرورت ارتباط بین گره ها ی دریافتی و انتقالی که انتقال واقعی را به انحصار در می اورد و به عنوان ارتباط RTS/CTS در نظر گرفته شده به طور کلی پذیرفته می شود در نمونه ارتباطی RTS/CTS (شکل ۳) یگ گره اماده انتقال به یک سبک می باشد و یک بسته کنترلی کوتاه را ارسال می کند REQUEST TO SEND (درخواست ارسال) با تمام گره هایی که تعویق RTS را از طریق کانال می شوند برای دوره ارتباط RTS/CTS می باشد مقصد به محض دریافت RTS با بسته های کنترلی کوتاه پاسخ می دهد تمام گره هایی که بسته CTS را می شنوند در طول انتقال بسته اطلاعاتی از دسترس به کانال عقب می افتند دریافت بسته CTS در انتقال گره تایید می کند که ارتباط RTS/CTS موفقیت امیز بوده و گره انتقال بسته اطلاعاتی واقعی را اغاز می کن در جایی که دو گره قالب های RTS همزمان را به گره مشابهی ارسال کنند انتقال دادهای RTS به هم برخورد کرده و از بین می روند اگر این اتفاق بیافتد گره هایی که بسته های RTS ناموفق را ارسال می کنند یک زمان سنج تصادفی را با الگوریتم تشریحی تنظیم می کنن اگر چه رابطه RTS/CTS مسئله های پایانه نمایان و مخفی را حذف می کند اما چند درجه پیشرفت را در طرح های CSMA سنتی فراهم می کند با این وجود رابطه ی RTS/CTS زمانی که گره ها دارای دامنه های انتقالی متفاوت می باشند از بین می رود پروتکل های MACAو 802/11 از رابطه RTS/CTS توصیف شده استفاده می کنند که برای اجتناب از برخورد روی کانال مشترک می باشد در جایی که IEEE802/11 متفاوت می باشد در نیاز ان برای تایید انتقال با دریافت کننده همراه بوده که پس از دریافت بسته اطلاعاتی انجام می گیرد گنجایش ACA در صورت نیاز انتقال خوری را باتغییر بسته دادها که به طور موفقیت امیزی دریافت شده اجرا می کند

۳-۱-۱. مثال اول :مسئله پایانه مخفی

در مثال زیر سه گره در دامنه ارتباطی وجود دارد گره صفر ارسال بسته های اطلاعاتی را به گره ۱ شروع می کند و پس از ان گره ۲ ارسال بسته های اطلاعاتی را به گره ۱ اغاز می کند پارامتر های ترکیب بندی برای این بخش به صورت زیر می باشد:

```
In nodes.input file:
                    0  0 (50,  1000, 0.0)
                    1  0 (650, 1000, 0.0)
                    2  0 (1250, 1000, 0.0)

so the simulation model is:
                Node0----600m----Node1----600m----Node2

In config.in file:
```

```
MOBILITY    =    NONE
SIMULATION-TIME   =    5S
PROPAGATION-PATHLOSS = FREE-SPACE
RADIO-TX-POWER   =   15dBm
RADIO-RX-SENSITIVITY =   -81 dBm
RADIO-RX-THRESHOLD   =   -81dBm
ANTENNA_GAIN   =   0dB
MAC_PROTOCOL   =   CSMA
```

در صورتی که فقط دو گره بسته های اطلاعاتی را به گره یک ارسال کنند در حالی که گره صفر به حال سکون بماند اما خروجی زیر را در فایل GLOMO.STAT بدست می اوریم :

IN APP.CONF:CBR 2 6 0572

```
Node: 1, Layer:        MacCSMA, Number of UNICAST packets received clearly: 1250
Node: 1, Layer:    AppCbrServer, (0) Total number of bytes received: 640000
Node: 1, Layer:    AppCbrServer, (0) Total number of packets received: 1250
Node: 2, Layer:        MacCSMA, Number of UNICAST packets output to the channel: 1250
Node: 2, Layer:    AppCbrClient, (0) Total number of bytes sent: 640000
Node: 2, Layer:    AppCbrClient, (0) Total number of packets sent: 1250
```

تعداد بسته های اطلاعاتی فرستاده شده (5S-0)/EEXP(-3)S=1250 یعنی تمام بسته هایی که با گره ۲ فرستاده شده اند توسط گره یک دریافت می شوند مانند همین مورد در صورتی که اجازه دهیم گره صفر به تنهایی منتقل شود و اجازه دهیم که هر دو گره صفر و۲ در زمانی مشابه مثل گره یک منتقل شوند خواهیم داشت :

```
In app.conf:    CBR 0  1   0    512   4MS  6MS  0
                CBR 2  1   0    512   4MS  0  0

Node: 0, Layer:        MacCSMA, Number of UNICAST packets output to the channel: 1248
Node: 0, Layer:    AppCbrClient, (0) Total number of bytes sent: 639488
Node: 0, Layer:    AppCbrClient, (0) Total number of packets sent: 1249
Node: 1, Layer:        MacCSMA, Number of UNICAST packets received clearly: 1605
```

```
Node: 1, Layer:    AppCbrServer, (0) Client address: 0
Node: 1, Layer:    AppCbrServer, (0) Total number of bytes received: 410624
Node: 1, Layer:    AppCbrServer, (0) Total number of packets received: 802
Node: 1, Layer:    AppCbrServer, (0) Client address: 2
Node: 1, Layer:    AppCbrServer, (0) Total number of bytes received: 411136
Node: 1, Layer:    AppCbrServer, (0) Total number of packets received: 803
Node: 2, Layer:         MacCSMA, Number of UNICAST packets output to the channel: 1250
Node: 2, Layer:    AppCbrClient, (0) Total number of bytes sent: 640000
Node: 2, Layer:    AppCbrClient, (0) Total number of packets sent: 1250
```

که نشان می دهد گره صفر ۱۲۴۸ بسته اطللاعاتی را ارسال می کند و گره ۲ ،۱۲۵۰ بسته اطلاعاتی اما گره یک فقط ۸۰۲ بسته اطلاعاتی را از گره صفر و ۸۰۳ بسته اطلاعاتی را از گره ۲ دریافت می کند در صورتی که سناریوی مشابهی راداشته باشیم و پروتکل MACA را تغییر دهیم نتایج زیر را خواهیم داشت :

```
In app.conf:    CBR 0  1   0    512   4MS  6MS  0
                CBR 2  1   0    512   4MS  0  0

Node: 0, Layer:         MacMACA, Number of UNICAST packets output to the channel: 162
Node: 0, Layer:         MacMACA, Number of RTS Packets sent: 4779
Node: 0, Layer:         MacMACA, Number of CTS Packets sent: 1
Node: 0, Layer:         MacMACA, Number of RTS Packets got: 1
Node: 0, Layer:         MacMACA, Number of CTS Packets got: 162
Node: 0, Layer:         MacMACA, Number of Noisy Packets got: 135
Node: 0, Layer:    AppCbrClient, (0) Server address: 1
Node: 0, Layer:    AppCbrClient, (0) Total number of bytes sent: 639488
Node: 0, Layer:    AppCbrClient, (0) Total number of packets sent: 1249
Node: 1, Layer:         MacMACA, Number of UNICAST packets received clearly: 281
Node: 1, Layer:         MacMACA, Number of RTS Packets sent: 40
Node: 1, Layer:         MacMACA, Number of CTS Packets sent: 2055
Node: 1, Layer:         MacMACA, Number of RTS Packets got: 2055
Node: 1, Layer:         MacMACA, Number of CTS Packets got: 2
```

```
Node: 1, Layer:    AppCbrServer, (0) Client address: 0
Node: 1, Layer:    AppCbrServer, (0) Total number of bytes received: 75264
Node: 1, Layer:    AppCbrServer, (0) Total number of packets received: 147
Node: 1, Layer:    AppCbrServer, (0) Client address: 2
Node: 1, Layer:    AppCbrServer, (0) Total number of bytes received: 68608
Node: 1, Layer:    AppCbrServer, (0) Total number of packets received: 134
Node: 2, Layer:         MacMACA, Number of UNICAST packets output to the channel: 144
Node: 2, Layer:         MacMACA, Number of RTS Packets sent: 4800
Node: 2, Layer:         MacMACA, Number of CTS Packets sent: 1
Node: 2, Layer:         MacMACA, Number of RTS Packets got: 1
Node: 2, Layer:         MacMACA, Number of CTS Packets got: 144
Node: 2, Layer:         MacMACA, Number of Noisy Packets got: 149
Node: 2, Layer:    AppCbrClient, (0) Total number of bytes sent: 640000
Node: 2, Layer:    AppCbrClient, (0) Total number of packets sent: 1250
```

و اگر ما به پروتکل IEEE802/11 تغییر دهیم خواهیم داشت

```
In app.conf:    CBR 0  1   0    512   4MS  6MS  0
                CBR 2  1   0    512   4MS  0  0

Node: 0, Layer:          802.11, UCAST (non-frag) pkts sent to chanl: 701
Node: 0, Layer:          802.11, retx pkts due to CTS timeout: 44
Node: 0, Layer:    AppCbrClient, (0) Total number of bytes sent: 639488
Node: 0, Layer:    AppCbrClient, (0) Total number of packets sent: 1249
Node: 1, Layer:          802.11, UCAST pkts rcvd clearly: 1425
Node: 1, Layer:    AppCbrServer, (0) Client address: 0
Node: 1, Layer:    AppCbrServer, (0) Total number of bytes received: 358912
Node: 1, Layer:    AppCbrServer, (0) Total number of packets received: 701
Node: 1, Layer:    AppCbrServer, (0) Client address: 2
Node: 1, Layer:    AppCbrServer, (0) Total number of bytes received: 370688
Node: 1, Layer:    AppCbrServer, (0) Total number of packets received: 724
Node: 2, Layer:          802.11, UCAST (non-frag) pkts sent to chanl: 724
Node: 2, Layer:          802.11, retx pkts due to CTS timeout: 44
Node: 2, Layer:    AppCbrClient, (0) Total number of bytes sent: 640000
Node: 2, Layer:    AppCbrClient, (0) Total number of packets sent: 1250
```

که نشان می دهد به هنگام استفاده از CSMA به جای MACA روش بهتری را به دست می اوریم که به خاطر پیام های RTS/CTS می باشد استفاده از بسته های اطلاعاتی RTS در هر موقعی که منبع دارای یک بسته اطلاعاتی باشد تا بدونه حساسیت اولیه کانال ارسال کن نتایج با تلاقی بسته ها و کاهش کلی بالا می رود مکانیزم پیشگیری از تلاقی به IEEE-802/*11 می پیوستد تا با انتقال بسته های FRTS به کاهش تعداد تلاقی ها کمک کند در نتیجه بیشتر بسته های اطلاعاتی به مقصد خود می رسند

۳-۱-۲.مثال دوم مسئله پایانه نمایان :

در مثال زیر چهار گره در دامنه ارتباطی هستند گره ۱ ارسال بسته های اطلاعاتی را یه گره صفر و پس از ان گره ۲ ارسال بسته های اطلاعاتی را به گره ۳ شروع می کند ما زمان حرکت درونی بسته را خیلی کوتاه کردیم به طوری که وقتی گره یک بخواهد انتقال یابد،مشغول بودن واسطهٔ اطلاعاتی را حس می کند.پارامترهای ترکیب بندی برای این بخش به صورت زیر است :

```
In nodes.input file:
                    0  0 (50,  1000, 0.0)
                    1  0 (650, 1000, 0.0)
                    2  0 (1250, 1000, 0.0)
                    3  0 (1850, 1000, 0.0)

so the simulation model is:
           Node0----600m----Node1----600m----Node2----600m----Node3

In config.in file:
                    MOBILITY    =  NONE
                    SIMULATION-TIME  =   5S
                    PROPAGATION-PATHLOSS = FREE-SPACE
                    RADIO-TX-POWER  =  15dBm
                    RADIO-RX-SENSITIVITY =  -81 dBm
                    RADIO-RX-THRESHOLD  =  -81dBm
                    ANTENNA_GAIN  =  0dB
                    MAC_PROTOCOL  =  CSMA
```

اگر ما اجازه دهیم که فقط گره ۲ ،بسته های اطلاعاتی را به گره ۳ ارسال کند ، در حالی که به حالت سکون باقی مانده ، بنابراین می توانیم خروجی را در فایل Glomo.stat بدست بیاوریم.

In app.conf: CBR 2 3 0 512 4MS 0 0

In glomo.stat:

Node: 2, Layer: MacCSMA, Number of UNICAST packets output to the channel: 1250

Node: 2, Layer: MacCSMA, Number of BROADCAST packets output to the channel: 1

Node: 2, Layer: AppCbrClient, (0) Total number of bytes sent: 640000

Node: 2, Layer: AppCbrClient, (0) Total number of packets sent: 1250

Node: 3, Layer: MacCSMA, Number of UNICAST packets received clearly: 1250

Node: 3, Layer: MacCSMA, Number of BROADCAST packets received clearly: 1

Node: 3, Layer: AppCbrServer, (0) Total number of bytes received: 640000

Node: 3, Layer: AppCbrServer, (0) Total number of packets received: 1250

تعداد بسته های اطلاعاتی که فرستاده شده اند با گره ۳ دریافت شده اند.همین مورد برای گره ۱ نیز رخ داده است که اگر گره ۱ به تنهایی انتقال یابد ما اجازه دهیم که گره ۱ و ۲ در زمان مشابهی انتقال یابند خواهیم داشت:

In app.conf:

CBR 1 0 0 512 4MS 4MS 0

CBR 2 3 0 512 4MS 0 0

In glomo.stat:

Node: 0, Layer: MacCSMA, Number of UNICAST packets received clearly: 0

Node: 0, Layer: MacCSMA, Number of BROADCAST packets received clearly: 0

Node: 1, Layer: MacCSMA, Number of UNICAST packets output to the channel: 0

Node: 1, Layer: MacCSMA, Number of BROADCAST packets output to the channel: 6

Node: 1, Layer: MacCSMA, Number of UNICAST packets received clearly: 0

Node: 1, Layer: MacCSMA, Number of BROADCAST packets received clearly: 1

Node: 1, Layer: AppCbrClient, (0) Total number of bytes sent: 639488

Node: 1, Layer: AppCbrClient, (0) Total number of packets sent: 1249

Node: 2, Layer: MacCSMA, Number of UNICAST packets output to the channel: 1250

Node: 2, Layer: MacCSMA, Number of BROADCAST packets output to the channel: 1

Node: 2, Layer: AppCbrClient, (0) Total number of packets sent: 1250

Node: 2, Layer: AppCbrClient, (0) Throughput (bits per second): 1024000

Node: 3, Layer: MacCSMA, Number of UNICAST packets received clearly: 1244

Node: 3, Layer: MacCSMA, Number of BROADCAST packets received clearly: 1

Node: 3, Layer: AppCbrServer, (0) Total number of bytes received: 636928

Node: 3, Layer: AppCbrServer, (0) Total number of packets received: 1244

که نشان می دهد انتقال گره ۲ به گره ۳ و تمام بسته های اطلاعاتی توسط گره ۲ دریافت شده اند ، گره ۱ کار انتقال را 4Ms پس از شروع شبیه سازی آغاز می کند، امّا هیچ حاملی را حس نمی کند.(اگر چه لایه کاربردی دارای ۱۲۴۹ بسته اطلاعاتی برای انتقال می باشد) و لایهٔ MACA تقریباًهیچ بسته ای را به کانال انتقال نمی دهد.

اگر ما پروتکل MAC را در فایل Config به MACA تغییر دهیم و اجازه دهیم که فقط گره ۲ بسته ها را به گره ۳ ارسال کند ، فایل Glomo.stat زیر را بدست می آوریم :

:In app.conf

CBR 2 3 0 512 4MS 0 0
Node: 2, Layer: MacMACA, Number of UNICAST packets output to the channel: 1220
Node: 2, Layer: MacMACA, Number of RTS Packets sent: 1220
Node: 2, Layer: MacMACA, Number of CTS Packets sent: 1
Node: 2, Layer: MacMACA, Number of RTS Packets got: 1

Node: 2, Layer: MacMACA, Number of CTS Packets got: 1220
Node: 2, Layer: AppCbrClient, (0) Total number of bytes sent: 624640
Node: 2, Layer: AppCbrClient, (0) Total number of packets sent: 1220
Node: 3, Layer: MacMACA, Number of UNICAST packets received clearly: 1219
Node: 3, Layer: MacMACA, Number of RTS Packets sent: 1
Node: 3, Layer: MacMACA, Number of CTS Packets sent: 1220
Node: 3, Layer: MacMACA, Number of RTS Packets got: 1220
Node: 3, Layer: MacMACA, Number of CTS Packets got: 1

که نشان می دهد تمام بسته های اطلاعاتی با گره ۲ فرستاده شده اند که توسط گره ۳ دریافت می شود . اگر ما اجازه دهیم گره ۱ به گره صفر و گره ۲ به گره ۳ به طور همزمان انتقال یابد آنگاه ما خواهیم داشت :

In app.conf:

CBR 1 0 0 512 4MS 4MS 0

CBR 2 3 0 512 4MS 0 0

Node: 0, Layer: MacMACA, Number of UNICAST packets received clearly: 41

Node: 0, Layer: MacMACA, Number of BROADCAST packets received clearly: 1

Node: 0, Layer: MacMACA, Number of RTS Packets sent: 64

Node: 0, Layer: MacMACA, Number of CTS Packets sent: 3162

Node: 0, Layer: MacMACA, Number of RTS Packets got: 3162

Node: 0, Layer: MacMACA, Number of CTS Packets got: 1

Node: 0, Layer: AppCbrServer, (0) Total number of bytes received: 20992

Node: 0, Layer: AppCbrServer, (0) Total number of packets received: 41

Node: 1, Layer: MacMACA, Number of UNICAST packets output to the channel: 721

Node: 1, Layer: MacMACA, Number of BROADCAST packets output to the channel: 1

Node: 1, Layer: MacMACA, Number of UNICAST packets received clearly: 1

Node: 1, Layer: MacMACA, Number of BROADCAST packets received clearly: 1

Node: 1, Layer: MacMACA, Number of RTS Packets sent: 3842

Node: 1, Layer: MacMACA, Number of CTS Packets sent: 21

Node: 1, Layer: MacMACA, Number of RTS Packets got: 21

Node: 1, Layer: MacMACA, Number of CTS Packets got: 721

Node: 1, Layer: MacMACA, Number of Noisy Packets got: 65

Node: 1, Layer: AppCbrClient, (0) Total number of bytes sent: 639488

Node: 1, Layer: AppCbrClient, (0) Total number of packets sent: 1249

Node: 2, Layer: MacMACA, Number of UNICAST packets output to the channel: 770

Node: 2, Layer: MacMACA, Number of BROADCAST packets output to the channel: 1

Node: 2, Layer: MacMACA, Number of UNICAST packets received clearly: 1

Node: 2, Layer: MacMACA, Number of BROADCAST packets received clearly: 0

Node: 2, Layer: MacMACA, Number of RTS Packets sent: 3837

Node: 2, Layer: MacMACA, Number of CTS Packets sent: 1

Node: 2, Layer: MacMACA, Number of RTS Packets got: 1

Node: 2, Layer: MacMACA, Number of CTS Packets got: 770

Node: 2, Layer: MacMACA, Number of Noisy Packets got: 42

Node: 2, Layer: AppCbrClient, (0) Total number of bytes sent: 640000

Node: 2, Layer: AppCbrClient, (0) Total number of packets sent: 1250

Node: 3, Layer: MacMACA, Number of UNICAST packets output to the channel: 1

Node: 3, Layer: MacMACA, Number of BROADCAST packets output to the channel: 0

Node: 3, Layer: MacMACA, Number of UNICAST packets received clearly: 52

Node: 3, Layer: MacMACA, Number of BROADCAST packets received clearly: 1

Node: 3, Layer: MacMACA, Number of RTS Packets sent: 1

Node: 3, Layer: MacMACA, Number of CTS Packets sent: 3179

Node: 3, Layer: MacMACA, Number of RTS Packets got: 3179

```
Node: 3, Layer: MacMACA, Number of CTS Packets got: 1
Node: 3, Layer: MacMACA, Number of Noisy Packets got: 0
Node: 3, Layer: AppCbrServer, (0) Total number of bytes received: 26624
Node: 3, Layer: AppCbrServer, (0) Total number of packets received: 52
```

که نشان می دهد در سناریوی پایانهٔ نمایان پروتوکل های MACA و CSMA عملکرد ضعیفی را نشان می دهند.

۳-۲ تجزیه و تحلیل نتایج :

اطلاعات جمع آوری شده در مثال های قبلی کم بود، به همین خاطر برای انجام یک تجزیه تحلیل مستقیم از فایل glomo.stat استفاده می کنیم . با این وجود در بشتر موارد ، توجهبه سایز شبکه وطول شبیه سازی امکان پذیر نمی باشد.در این مورد ما از ابزارهای دیگر برای انجام تجزیه و تحلیل از فایل glomo.stat استفاده می کنیم ،برای مثال در برنامه نویسی زیر داریم :

```
./#analy-datpak1.sh <file>
cat $1 | grep AppCbrServer | grep Total | grep packets | grep received
```

که تعداد بسته های اطلاعاتی دریافت شده را بدست می آورد.یک نمونه از این مورد مه با فایل glomo.stat اجرا می شود و به صورت زیر است :

```
./analy-datpak1.sh cmpmac-802aodv-200.stat
Node: 3, Layer: AppCbrServer, (0) Total number of packets received: 404
Node: 5, Layer: AppCbrServer, (0) Total number of packets received: 1115
Node: 7, Layer: AppCbrServer, (0) Total number of packets received: 622
Node: 9, Layer: AppCbrServer, (0) Total number of packets received: 260
```

```
Node: 62, Layer: AppCbrServer, (0) Total number of packets received: 360
Node: 65, Layer: AppCbrServer, (0) Total number of packets received: 338
Node: 67, Layer: AppCbrServer, (0) Total number of packets received: 779
Node: 69, Layer: AppCbrServer, (1) Total number of packets received: 1011
Node: 70, Layer: AppCbrServer, (0) Total number of packets received: 922
Node: 73, Layer: AppCbrServer, (0) Total number of packets received: 1079
Node: 76, Layer: AppCbrServer, (0) Total number of packets received: 935
Node: 79, Layer: AppCbrServer, (0) Total number of packets received: 939
Node: 81, Layer: AppCbrServer, (0) Total number of packets received: 647
Node: 83, Layer: AppCbrServer, (0) Total number of packets received: 632
Node: 85, Layer: AppCbrServer, (0) Total number of packets received: 1184
Node: 88, Layer: AppCbrServer, (0) Total number of packets received: 875
Node: 90, Layer: AppCbrServer, (0) Total number of packets received: 988
Node: 94, Layer: AppCbrServer, (0) Total number of packets received: 363
Node: 95, Layer: AppCbrServer, (0) Total number of packets received: 1014
Node: 96, Layer: AppCbrServer, (0) Total number of packets received: 874
```

پارامتر های دیگر مانند Packet Data Ratio نسبت اطلاعات بسته Percentage درصد از میزان اتلاف و بسته های فرستاده شده توسط منابع می تواند با برنامه نویسی awk زیر به دست آید :

Shell program:

```
cat $1 | grep App | grep Total | grep packets | awk -f glomopak1.awk
```

AWK program:

```
BEGIN{
sumcountsent = 0;
```

```
sumcountrcv = 0;
}
{
if ($10 == "sent:") sumcountsent+= $11;
else if ($10 == "received:") sumcountrcv += $11;
}
END{
printf("Loss Packet Percentage = %f \% \n", 100-((sumcountrcv*100)/sumcountsent));
printf("Packet Delivery Ratio = %f \n\n",sumcountrcv/sumcountsent);
printf("Packets Received = %d \n", sumcountrcv);
printf("Packets Sent = %d \n\n", sumcountsent);
}
```

یک مثال از اجرای چنین برنامه هایی که در مثال قبل هم بیان شد بصورت زیر است :

```
./analy-datpak2.sh cmpmac-802aodv-200.stat
Loss Packet Percentage = 36.079167 %
Packet Delivery Ratio = 0.639208
Packets Received = 15341
Packets Sent = 24000
```

برای بدست میانگین تحویل (در مورد ترافیک CBR) وکنترل بسته های جمع آوری شده توسط پروتکل AODA از برنامه نویسی زیر استفاده می شود :

```
# program analy-delay1.sh
cat $1 | grep AppCbrServer | grep end-to-end | grep delay | awk -f delay1.awk
where delay1 AWK program is:
BEGIN{
```

```
sumdelay = 0;
countdelay = 0;
}
{
sumdelay += $10;
countdelay++;
}
END{
printf("Average Delay = %f \n", sumdelay/countdelay);
}
```

```
# program analy-routAODVctrl-1.sh
cat $1 | grep RoutingAodv | grep CTRL \
| awk '{sum += $11} END {print sum}'
```

ما می توانی برخی گراف های جالب را از شبیه سازی بدست آوریم . فرض کنید می خواهیم گرافی از واریانس بسته های تحویل داده شده به مقصد نهایی را به دست آوریم که دارای سایزهای مختلفی از شبکه ها (مانند ۳۰ ، ۵۰ ، ۷۰، ۱۰۰، ۲۰۰ گره) سرعت های مختلفی از گره ها می باشد (۱ ، ۵، ۱۰ ، ۲۰m/s) . ما می توانیم چندین شبه سازی را انجام دهیم (که هرکدام با تعداد گره مشخص و سرعت مدل سیّار می باشند) . آزمایشات را می توان برای کاهش زمان شبیه سازی ، در زمان مشابهی اجرا کرد ، بنابراین جدول زیر بدست

می آید :

Speed	30 nodes	50 nodes	70 nodes	100 nodes	200 nodes
1	12885	34940	56539	57514	67310
5	16927	33433	43038	36315	68720
10	14393	22421	34559	34884	44615
20	12108	20455	29128	27706	28297

در صورتی که داده های این جدول در یک فایل استفاده شوند (برای مثال Mobil.txt) آنگاه ممکن است که از ابزار گرافیکی مثل GnuPlot برای بدست آوردن گرافی مناسب استفاده کرد . در مورد GnuPlot می توان فرمان های زیر را برای بدست آوردن گراف ارئه شده در شکل ۴ اجرا کرد :

```
set terminal postscript eps
set size 1/1., 1/1.
```

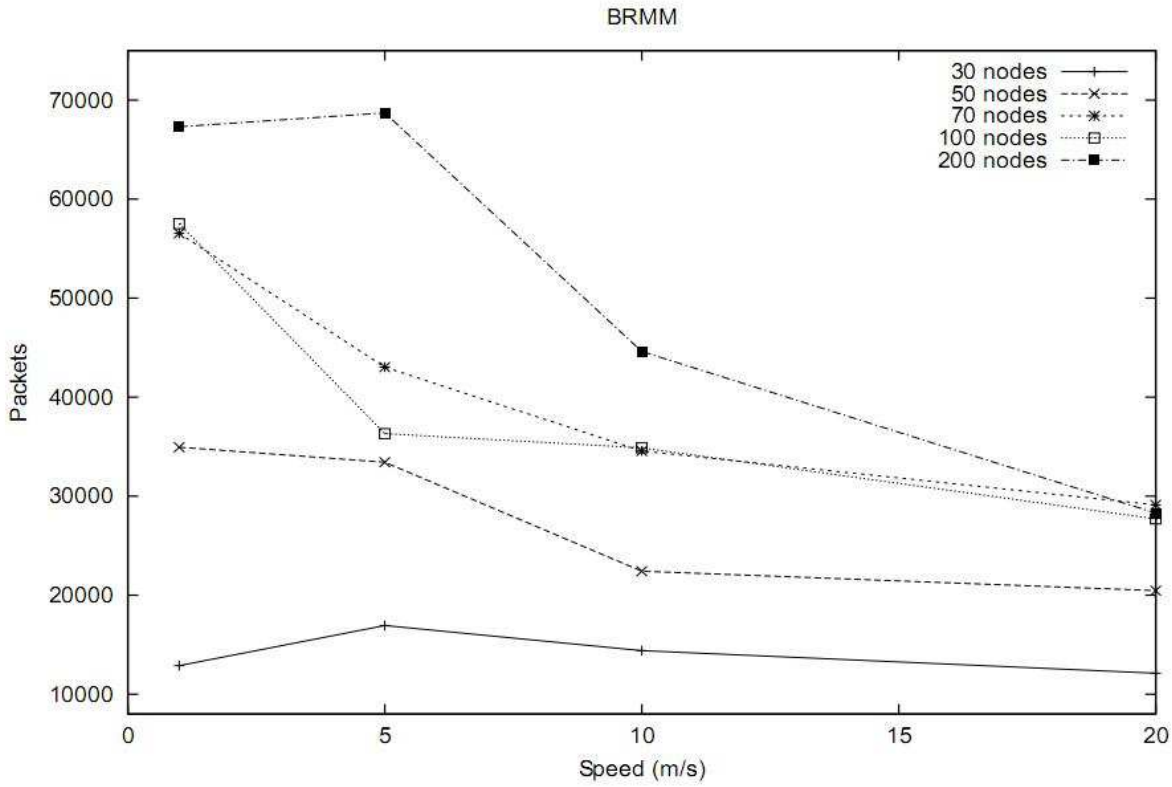


Figure 4: Number of Data Packets received by their destination

```
set title "BRMM"
set output "nodedens12.eps"
set xlabel "Speed (m/s)"
```

```
set ylabel "Packets"
plot [0:20][8000:75000] "mobil22.txt" using 1:2 title '30 nodes' with linespoints, \
"mobil.txt" using 1:3 title '50 nodes' with linespoints, \
"mobil.txt" using 1:4 title '70 nodes' with linespoints, \
"mobil.txt" using 1:5 title '100 nodes' with linespoints, \
"mobil.txt" using 1:6 title '200 nodes' with linespoints
```

۴- گره پروتوکل های مسیریابی در Glomosim

۴-۱ AODV

Glomosim بر اساس یک پیش نویس IETF قدیمی تر،ADOV را اجرا می کند .فرض بر این است که این پروتوکل، پروتکل MAC ، سیگنالی را برای پروتکل مسیریابی ارسال می کند. این زمانی اتفاق می افتد که این پروتکل ارتباط را قطع می کند. پروتکل های MAC مثل IEEE802.11 و MACAW همچنین کاربردی را دارند.برای مثال در EEE802.11 هنگامی که هیچ CTS پس از RTS دریافت نمی شود و هیچ ACK پس از انتقال بسته ها دریافت نمی شود ، سیگنال به پروتکل مسیریابی ارسال می شود .

اگر کاربران بخواهند از پروتکل های MAC به جای IEEE802.11 استفاده کنند ، می بایست طرح ها را برای کشف قطع ازتباط اجرا کنند . یک روش برای انجام این کار استفاده از بسته های اطلاعاتی HELLO است که در سند های ADOV مشخص شده اند. RREPهای ناخواسته ، منتشر می شوند و تنها در صورتی پیش می رود که گره ، بخشی از مسیر شکسته شده باشد ، نه منبع آن مسیر. اگر بیشتر از یک مشیر از پیوند نا موفق استفاده کند ، پیام های چندگانه RREP ارسال می شوند.

۴-۲ FSR

FSR بر اساس این فرض است که تغییرات پیکر بندی شبکه ، تاثیر کمی روی بسته اطلاعاتی مسیر یاب دارد فاصله بین مسیر یاب و شبکه را افزایش می دهد. FSR تطبیقی از GSR است که به جای اطلاعات انتشار یافته از طریق شبکه و تغییرات دوره ای ، گره ، وضعیت جدول ارتباطی فردی را نسبت های متفاوت و بسته به فاصلهٔ منبع پیوند آغاز می کند . هر پیام به روز اطلاعاتی را در مورد تمام گره ها در بر دارند . این پیام ، اطلاعات را در موردگره های نزدیک تر مبادله می کند که که متداول تری گره های دورتر بوده و بنابراین سایز پیام به روز را کاهش می

دهد . شکل ۵ محدودهٔ fisheye را برای کرهٔ مرکزی تعریف می کند که مجموعه ای از گره هاست که در چار چوب مسیر انتقال موج به هم می رسند. در گلوموسیم

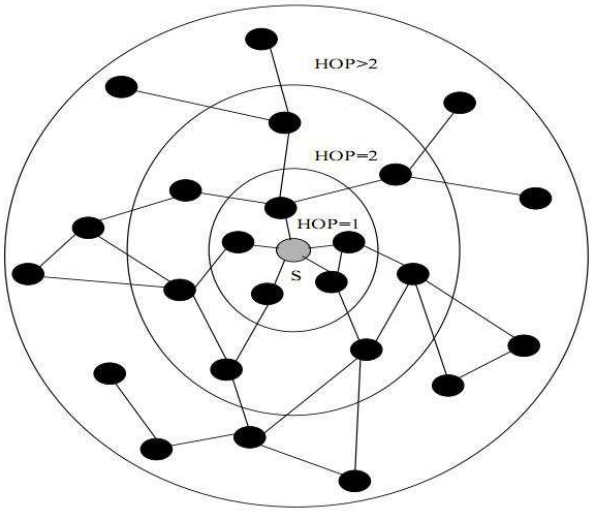


Figure 5: Accuracy of information in FSR

وقتی که FSR به عنوان پروتکل مسیریابی استفاده می شود ، ورودی های زیر را می بایست در فایل Config.in تعریف کرد :

ROUTING-PROTOCOL FISHEYE

FISHEYE-FILE FISHEYEConfig

فایل fisheyeConf شامل پارامتر های مسیر یاب زیر می باشد :

- اندازهٔ صفحهٔ رادار : این پارامترهای مسیر یاب زیر می باشد :
- وقفه برای گره های هم جوار : اگر یک گره از گره مجاوری که با این مقدار مشخص شده است ، فرمانی نشنود ، گره هم جوار از فهرست هم جواری حذف می شود .
- فاصلهٔ بهنگام سازی صفحهٔ رادار درونی : فاصله بهنگام سازی ارسال و بهنگام سازی گره ها در شعاع های صفحه رادار

یک نمونه از سناریو با استفاده از پروتکل مسیر یاب FSR را می توان در فهرست راهنمای:

scenarios/ routing-fisheye/glomosim-2.03/ glomosim/ یافت.

۳-۴ WRP

پروتکل مسیریابی بی سیمی (WRP) یک پروتکل پیش فعال است که اطلاعات مسیریابی را از طریق مبادلهٔ بهنگام سازی دورهای حفظ می کند. گره ای که به طور موفقیت آمیزی پیام به روز رسانی را دریافت می کند ، فرمانتایید را به فرستنده انتقال می دهد پیوند هنوز قابل دوام است . در رویدادی که یک گره چیزی را در دورهٔ زمانی مشخص شده انتقال نمی دهد ، می بایست پیام HELLO را منتقل کرد تا از اتصال آن اطمینان حاصل شود . در غیر این صورت ، فقدان پیام از یک گره ، شکست آن پیوند را نشان می دهد . هنگامی که یک گره ، پیام HELLO را از گره جدید دریافت می کند ، نسخه ای از اطلاعات جدول مسیر یابی را ارسال می کند . هر گره ، جدول فاصله ،مسیر یابی جدول هزینهٔ پیوند و فهرست انتقال پیام را حفظ می کند . اجرای WRP در Glomosim بر اساس شبکه کد می باشد که در بخش ۱۱ به آن اشاره شده است .

با این وجود تعداد نسبتاً زیادی از جداول که این پروتکل نیاز به حافظه سختی دارد بر روی دستگاه های سیّار قرار دارند خصوصاً وقتی که شبکه شروع به رشد و ترقی می کند . که در عوض با توان باطری دستگاه سیّار، تقاضاهای بیشتری را می طلبد ، و در مورد Glomosim شبیه سازی های بیش از صد گره در خطای هستهٔ تجزیه ، رخ می دهد در چنین مواردی پارامتر Number-Of-NODES را می بایس تا ۹۹ تنظیم کرد.

۵ – راهنمای طراح نرم افزار Glomosim :

۵-۱ کد گزاری در Glomosim

در بسیاری از موارد ما علاقه داریم که فایل های منبع را تغییر دهیم و یا فایل های جدید را به منظور اضافه کردن یا بدست آوردن کاربردهای جدید ایجاد کنیم. پروتوکل های جدید را می توان با GLOMOSIM در هر لایه از مدل OSI نوشت و به طور کلی سبک کد گذاری فایل منبع را می بایست زمان تغییر کدگذاری، دنبال کرد. برخی از توصیه های کلی در یزر آورده شده است:

- از فرآیندهای توصیفی و قابل خواندن و اسامی قابل دسترس استفاده کنید. از کلمات اختصاری یا رمز بندی ها استفاده نکنید. از کلمات واقعی اسامی مثل Header استفاده کنید نه Hdr.
- همخوان های ماکرو می بایست بالاتر از حروفی که زیر آنها خط کشیده باشند.
- شناسه های عملکرد می بایست شامل یک فعل باشد، برای مثال می توان به مقدار دهی مقادیر (Initialize value) و Create Qued اشاره کرد.
- از بیشتر از ۸۰ ستون استفاده نکنید، چون خواندن خطوط خیلی دشوار می باشد.
- فرمان های بلوک، خطوط جدا از کد می باشند. آنها برای سطح مشابهی که کدها دارند، فاصله گذاشته می شود.
- قالب بندی فرمان های بلوک به صورت زیر می باشد:

```
//
// This is a block comment. First line.
// This is a block comment. Second line.
// This is a block comment. Third line.
// This is a block comment. Fourth line.
//
or
/*
* This is a block comment. First line.
* This is a block comment. Second line.
* This is a block comment. Third line.
* This is a block comment. Fourth line.
*/
```

فایل های منبع می بایست طرح زیر را داشته باشند: می بایست فرمان بلوک، نام فایل، هدف آن و در صورت امکان تاریخچهٔ چاپ آن را ارائه دهد. برای فایل های سرآمد می بایست همیشه از بلوک # infndef HEADERFILE-H ، # define HEADERFILE-H ، # endif استفاده کرد. فایل مرجع را می بایست با ترتیبی منطقی سازماندهی

کرد. کاربردها، گونه ها و دستورالعمل هایی که فقط در فایل تکی استفاده می شوند به نام ایستا معرفی می شوند و در فایل مرجع قرار داده می شوند نه در فایل سرآمد. کاربردها، گونه ها و دستورالعمل هایی که با فایل هایی دیگر مرجع استفاده می شوند، در یک فایل سرآمد معرفی می شوند. برای توضیح کاربرد قبل از هر چیزی، نمونهٔ اولیه را با یک فرمان بلوک که شامل اسم کاربرد، توصیف هدف آن، فرضیات مربوط به آن می باشد، نشان می دهند. شما ممکن است هر بخشی را و تأثیر آن را فهرست بندی کنید، هر چند که از توصیف کردن واضح تر می باشد. برای مثال:

```
//
// FUNCTION: IsEmptyQueue()
// PURPOSE: Checks if the queue is empty.
// RETURN VALUE: Returns True if the queue is empty
// ASSUMPTIONS: None.
//
```

هر جزء از کدهای تو در تو را با چهار فاصله می نویسند. کاراکترهای TBA با فرمان expand تغییر می یابند. کاراکترهای TBA با فایل emacs کاهش می یابند. پیشنهادات دیگری وجود دارد که در زیر به برخی از آنها اشاره شده است:

هر جزء از کدهای تو در تو را با چهار فاصله می نویسند. کاراکترهای TBA با فرمان expand تغییر می یابند. کاراکترهای TBA با فایل emacs کاهش می یابند. پیشنهادات دیگری وجود دارد که در زیر به برخی از آنها اشاره شده است:

- از همخوان های موجود برای ارائه اسامی به مجموعه ای از پرچم ها استفاده کنید.

```
if (.....) {
Statement
}
else {
Statement
}
```

- کاربردهایی مثل( ) bcopy، ( ) bzero، کاربدهای استاندارد غیر ANSI می باشند و بنا بر مورد استفاده استاندارد ANSI هستند.

```
BAD:
#define  INTEGER    0
#define  CHAR       1
#define  FLOAT      2
#define  DOUBLE     3
```

```
GOOD:
typedef enum {
    Integer,
    Char,
    Float,
    Double
} DataType;
```

۵-۲ ساختار GLOMOSIM

پس از دانلود کردن GLOMOSIM، دستورالعمل های زیر به کار می رود:

application/ حاوی کدی برای سطح کاربرد

bin/ برای فایل های ورودی یا خروجی و قابل اجرا

doc/ حاوی اسناد

include/ حاوی فایل های معمولی

java- gui/ حاوی ابزار حسابرسی

Mac/ حاوی کدی برای سطح mac

Main/ حاوی طرح چارچوب اساسی

network/ حاوی کدی برای سطح شبکه

radio/ حاوی کدی برای سطح رادیویی

scenarios/ حاوی برخی سناریوهای نمونه

tc plic/ حاوی کتابخانه هایی برای TCP

transport/ حاوی کدی برای سطح انتقال

فهرست راهنمای اصلی شامل چارچوبی اساسی برای اجرای GLOMOSIM می باشد که شامل فایل driver.pc است که ورودی راه انداز را تعریف می کند. هر برنامهٔ parsec می بایست شامل ورودی به نام driver باشد که هدف مشابهی را با کاربرد اصلی برنامهٔ c دارد. این فایل، فایل ترکیب بندی را می خواند، شبیه سازی را آغاز می کند و نتایج آمار نهایی فایل glomo.stet از چند فایل موقت، می نویسد. نتایج چنین رویدادی در زمان اجرا به صورت زیر می باشد:

۱- کاربرد اصلی در driver.pc اجرا می شود که در محل شروع GLOMOSIM کاربرد اصلی C محسوب می شود.

۲- کاربرد اصلی، parsec.main نامیده می شود که شبیه سازی parsec را شروع می کند و ورودی driver را ایجاد می کند. کاربرد parsec.main زمانی استفاده می شود که کاربر بخواهد main خودش را بنویسد و در pcc- DIRECTORY/include/pc.api پیدا کند (چون این کاربر بخشی از سیستم اجراسازیparsec می باشد، دسترسی به مرجع آن امکان پذیر نمی باشد.).

۳- وقتی شبیه سازی پایان می یابد، parsec-main بر می گردد و مابقی کاربرد اصلی اجرا می شود. دلیل اینکه GLOMOSIM از کاربرد اصلی استفاده می کند این است که می بایست آمار ما را جمع آوری کرده و به فایل واحدی انتقال دهد و فایل را ببندند.

بنابراین کاربردی که پس از تمام ورودی های glomo اجرا می شوند، به پایان می رسند. یک نمونه از چگونگی استفاده از کاربرد parsec-main برای پرنیت یک پیام در برنامه hola.pc زیر ارائه شده است.

```
#include <stdio.h>

int main(int argc, char **argv){

parsec_main(argc,argv);

printf("Simulation Ended\n\n");

}

entity driver (int argc, char **argv) {

int i;

printf("HELLO WORLD \n\n");

}
```

هویت driver یا کارانداز، کاربرد ()parsec-main نامیده می شود که پیام HELLO WORLD را چاپ می کند، آنگاه شبیه سازی پایان می یابد و پیام «Simulation Ended» چاپ می شود. فرمان ها برای کامپایل کردن برنامه به صورت زیر نشان داده شده است. گزینهٔ clock نوع ساعت را برای استفاده مشخص می کند. در حالی که گزینهٔ user-main، اسم ( ) rename main را به ( )parsec-main برای یک user- defined main تغییر می دهد.

```
> pcc -g -O3 -clock longlong -lm -c hola.pc
> pcc -g -O3 -clock longlong -user_main hola.o -lm -o hola
>./hola
```

```
HELLO WORLD
Execution time : 0.0007 sec
Number of events (including timeouts) processed : 0
Number of messages processed : 0
Number of context switches occurred : 6
Number of Local NULL messages sent : 0
Number of Remote NULL messages sent : 0
Total Number of NULL messages sent : 0
Simulation Ended
>
```

در GLOMOSIM، هویت DRIVER یا راه انداز (in/main/driver.pc) شناسهٔ فایل خروجی را می خواند، بخش بندی هایی را ایجاد می کند، حافظه را برای اطلاعات گره تخصیص می دهد و کاربردهای مناسبی وابسته به مقادیر خروجی خواندنی مانند به جای ورودی بخش GLOMOsim (که در فایل glomo.pc تعریف شده است) شبیه سازی را آغاز می کند. فایل main/glomo.pc/. مراحل زیر را انجام می دهد:

۱- اطلاعات را از ورودی راه انداز دریافت می کند.

۲- گره هایی که متعلق به بخش قبلی بوده را پیدا می کند و تمام لایه ها را برای این گره ها، با فراخوانی هویت های GLOMO- ، GLOMO-Macinit () ، GLOMO-RadioIinit() ، GLOMO-propinit () Netwarklint ، GLOMO-TransportInit () ،GLOMO-APPiNIT () ، GLOMO-Mobility () مقدار دهی می شوند.

۳- به حلقهٔ تکرار بروید و سعی کنید پیام ها را دریافت کنید. وقتی پیامی دریافت می شود، اطلاعات در مورد دریافت گره و فراخوانی لایهٔ مناسب یافت می شود. این پیام زمان شبیه سازی سابق را در طول اجرای برنامه، نشان می دهد.

۴- هنگامی که شبیه سازی پایان می یابد به مرحلهٔ نهایی کدگذاری می رسد. این کد، کاربرد نهایی را برای تمام لایه های گره ها در این بخش، فرا می خواند. طراح نرم افزار می تواند آمارهای مورد نیاز را جمع آوری کند.

کاربردهای نهایی عبارتند از: () GLOMO-APPFinalize ، GLOMO-MACFinalize ، GLOMO-Networkcfinalizer ، GLOMO-Transportfinalize ، GLOMO-APPFinalize و GLOMO-Mobilityfinalize.

کاربردهای گفته شده تمام لایه ها را مقدار دهی، فراخوانی و به پایان می رساند که در فایل /include/struotmsg.h تعریف شده و در فهرست راهنمای مناسبی کد گذاری می شود. برای مثال() GLOMO-RadioInit در /radio/radio.pc کد گذاری می شود، در حالی که () GLOMO-MacInit ، در /mac/mac.pc کد گذاری در /transport/transport.pc کد گذاری می شدند.

۳-۵- طراحی GLOMOSIM

GLOMOSIM به روشی لایه ای و با استاندارد APIs طراحی شده است که بین لایه های شبیه سازی متفاوت استفاده می شود. انبارهٔ حافظه پروتوکل شامل مدلهایی برای کانال، رادیو، MAC ، شبکه، حمل و نقل و لایه های بالاتر می باشد. API های هستهٔ مرکزی GLOMOSIM به شکل فراخوانی های کاربردی می باشند، در حالی که

برای لایه های دیگر، API به شکل پیام بوده و با لایه ها مبادله می شوند. چندین API در بقیهٔ این بخش ارائه شده است.

۵-۳-۱ زمان سنج ها:

GLOMOSIM دارای چندین زمان سنج ساختگی در لایه های متفاوت می باشد. به طور کلی یک زمان سنج، رویداری که با لایه یا سطح تطابق همراه است را برنامه ریزی می کند. فایل glomosim/include/structmsg.h/ رویدارهایی را تعریف می کند که در GLOMOSIM کاربرد دارند، اگر چه کاربر می تواند رویدادهای تعریف شدهٔ خودش را به فایل اضافه کند.

```
/* Message Types for Channel layer */
MSG_CHANNEL_FromChannel,
MSG_CHANNEL_FromRadio,
/* Message Types for Network layer */
MSG_NETWORK_FromApp,
MSG_NETWORK_FromMac,
MSG_NETWORK_FromTransportOrRoutingProtocol,
MSG_NETWORK_DelayedSendToMac,
MSG_NETWORK_RTBroadcastAlarm,
MSG_NETWORK_CheckTimeoutAlarm,
MSG_NETWORK_TriggerUpdateAlarm,
MSG_NETWORK_InitiateSend,
MSG_NETWORK_FlushTables,
MSG_NETWORK_CheckAcked,
MSG_NETWORK_CheckReplied,
```

ما می توانیم پروتوکل مسیریابی AODV را به عنوان یک مثال در نظر بگیریم. هنگامی که در پروتوکل، یک گره، درخواست مسیریابی را ارسال می کند (RREQ)، می بایست پاسخ مسیریابی را با استفاده از زمان سنج دریافت کرد. بنابراین در فایل /glomosim/network/adov.h تعریف شده اند)، آخرین دستور العمل کاربر Routing RREQ AddvInitiale به صورت زیر می باشد:

```
void RoutingAodvInitiateRREQ(GlomoNode *node, NODE_ADDR destAddr)
{
. . .
RoutingAodvSetTimer(node, MSG_NETWORK_CheckReplied, destAddr,
(clocktype)2 * ttl * NODE_TRAVERSAL_TIME);
} /* RoutingAodvInitiateRREQ */
```

کاربرد Routing Aodvset Timer زمان سنج را با ارسال یک پیام به سطح شبکه تنظیم می کند. آخرین بحث در مورد این کاربرد، تأخیر آن از زمان شبیه سازی اخیر می باشد. NODE-TRAVERSAL-TIME با پیش نویس EOMS در فایل /glomosim/network/adov.h تعریف می شود. Function.Rauting AodvsetTimer به صورت زیر تعریف می شود:

```
void RoutingAodvSetTimer(
GlomoNode *node, long eventType, NODE_ADDR destAddr, clocktype delay)
{
Message *newMsg;
```

```
NODE_ADDR *info;

newMsg = GLOMO_MsgAlloc(node,

GLOMO_NETWORK_LAYER,

ROUTING_PROTOCOL_AODV,

eventType);

GLOMO_MsgInfoAlloc(node, newMsg, sizeof(NODE_ADDR));

info = (NODE_ADDR *) GLOMO_MsgReturnInfo(newMsg);

*info = destAddr;

GLOMO_MsgSend(node, newMsg, delay);

} /* RoutingAodvSetTimer */
```

هنگامی که زمان سنج خاتمه می یابد، کاربرد Routing Aodv Handle Protocol Event عملکرد مناسبی را برای این رویداد به وجود می آورد.

```
void RoutingAodvHandleProtocolEvent(GlomoNode *node, Message *msg)

{

. . .

switch (msg->eventType) {

. . .

/* Check if RREP is received after sending RREQ */

case MSG_NETWORK_CheckReplied: {
```

```
NODE_ADDR *destAddr = (NODE_ADDR *)GLOMO_MsgReturnInfo(msg);

/* Route has not been obtained */

  if (!RoutingAodvCheckRouteExist(*destAddr, &aodv->routeTable))

  {

  if (RoutingAodvGetTimes(*destAddr, &aodv->sent) < RREQ_RETRIES)

  {

  /* Retry with increased TTL */

  RoutingAodvRetryRREQ(node, *destAddr);

  } /* if under the retry limit */

          . . .

} /* RoutingAodvHandleProtocolEvent */
```

٥-٤ چگونگی اضافه کردن پروتوکل جدید:

به هنگام اضافه کردن یک مدل یا پروتوکل به یک سطح به منظور حفظ ثبات و کاربرد مناسب در GLOMOSIM، گام های متفاوتی را می بایست در نظر گرفت. سه کاربرد اصلی برای اضافه کردن یک پروتوکل جدید در ذیل شرح داده شده است.

۱- کاربرد ارزش دهی آغازی یا فرمت، این کاربرد داده های ویژهٔ مدل را تخصیص داده و فرمت می کند.

۲- کاربرد به پایان رسانی: این کاربرد آمارهای خروجی را از اجرای شبیه سازی، برای این مدل ایجاد می کند.

۳- کاربرد رویداد شبیه سازی: این کاربرد عملکردهای شبیه سازی را زمانی انجام می دهد که با یک رویداد برنامه ریزی شده باشد. ما از پروتوکل جدیدی به نام NEWPROT استفاده می کنیم تا مراحل اصلی را برای تعریف یک پروتوکل در شبکه، نشان دهیم. این پروتوکل جدید، پروتوکل مسیریابی AODV نامیده می شود که به منظور ساده سازی توضیحات و نشان دادن مکانیزم مورد استفاده در GLOMOSIM می باشد و مدل جدیدی را ارائه می دهد.

ما با نشان دادن چگونگی نامیدن سطح شبکه، آغاز می کنیم. همانطور که قبلاً بیان شد فایل glomo.pc اطلاعاتی را از ورودی راه انداز دریافت می کند و تمام سطح ها را برای گره ها فرمت کرده و به حلقهٔ نامتناهی برای دریافت پیام می رود. هنگامی که یک پیام دریافت می شود، گره و لایهٔ مناسبی را با کاربرد ( ) GLOMO-calllayer فرا می خواند:

```
GLOMO_CallLayer(GlomoNode *node, Message * msg){
. . .
switch (GLOMO_MsgGetLayer(msg) ){
case GLOMO_RADIO_LAYER:
GLOMO_RadioLayer(node, msg);
break;
case GLOMO_MAC_LAYER:
GLOMO_MacLayer(node, msg);
break;
case GLOMO_NETWORK_LAYER:
GLOMO_NetworkLayer(node, msg);
break;
case GLOMO_TRANSPORT_LAYER:
```

```
GLOMO_TransportLayer(node, msg);
break;
case GLOMO_APP_LAYER:
GLOMO_AppLayer(node, msg);
break;
}
. . .
}
```

کاربرد ( ) Glomo-Network Layer رفتار سطح شبکه را برای دریافت پیام، مدلسازی می کند و کاربرد Network ( )Iplayer را که در فایل ./network/nwip.pc تعریف شده است، فرا می خواند. اولین مرحله در تعریف مسیریابی ارزش دهی آغازی در فایل، شامل می شود. در این مثال کد ضروری را به منظور تشخیص پروتوکل NEWPROT اضافه می کنیم:

```
void NetworkIpInit(GlomoNode* node, const GlomoNodeInput* nodeInput)
{
. . .
retVal = GLOMO_ReadString(node->nodeAddr, nodeInput,
"ROUTING-PROTOCOL", protocolString);
if (retVal == FALSE) {
printf("CONFIG.IN Error: ROUTING-PROTOCOL not specified!\n");
assert(FALSE); abort();
}
```

```
. . .
else if (strcmp(protocolString, "AODV") == 0) {
ipLayer->routingProtocolChoice = ROUTING_PROTOCOL_AODV;
RoutingAodvInit(
node, (GlomoRoutingAodv**)&ipLayer->routingProtocol, nodeInput);
}
/* ******* THIS IS THE ADDED CODE ******* */
else if (strcmp(protocolString, "NEWPROT") == 0) {
ipLayer->routingProtocolChoice = ROUTING_PROTOCOL_NEWPROT;
RoutingAodvInit(
node, (GlomoRoutingAodv**)&ipLayer->routingProtocol, nodeInput);
}
. . .
}
```

این کاربرد(تابع)، مقدار خواندن را از فایل ترکیب بندی خروجی می گیرد و مسیریابی ارزش دهی آغازی مناسبی را فرا می خواند. همچنین مسیریابی پایانی را در فایل nwip.pc تعریف می کنند:

```
void NetworkIpFinalize(GlomoNode *node)
{
GlomoNetworkIp* ipLayer = (GlomoNetworkIp*)node->netwo
```

```
switch (ipLayer->routingProtocolChoice) {
case ROUTING_PROTOCOL_LAR1:
NetworkLar1Finalize(node);
break;
case ROUTING_PROTOCOL_AODV:
RoutingAodvFinalize(node);
break;
case ROUTING_PROTOCOL_NEWPROT:
RoutingAodvFinalize(node);
break;
. . .
}
```

ROUTING-PROTOCOL-NEWPROT در این کاربردها استفاده می شود که می بایست در فایل /include/network.h بیان شود.

```
typedef enum {
NETWORK_PROTOCOL_IP = 0,
ROUTING_PROTOCOL_AODV,
ROUTING_PROTOCOL_NEWPROT, /* New Protocol added */
ROUTING_PROTOCOL_DSR,
ROUTING_PROTOCOL_LAR1,
```

```
ROUTING_PROTOCOL_ODMRP,
ROUTING_PROTOCOL_OSPF,
ROUTING_PROTOCOL_ZRP,
ROUTING_PROTOCOL_ALL,
ROUTING_PROTOCOL_NONE
} NetworkRoutingProtocolType;
```

آنگاه کاربرد اداره کردن رویداد شبیه سازی می بایست تغییر کند، در این مورد () NetworkIplayer می تواند در فایل nwip.pc یافت شود:

```
void NetworkIpLayer(GlomoNode *node, Message *msg) {
switch (msg->protocolType) {
. . .
case ROUTING_PROTOCOL_AODV: {
RoutingAodvHandleProtocolEvent(node, msg);
break;
}
case ROUTING_PROTOCOL_NEWPROT: {
RoutingAodvHandleProtocolEvent(node, msg);
break;
}
. . .
```

پروتوکل اینترنتی دارای یک فیلد ۸ بیتی است که پروتوکل (IPVE) یا Next Header نامیده می شود که پروتوکل سطح جدید را شناسایی می کند. مرحله بعد، تعریف شمارهٔ پروتوکل برای پروتوکل جدید ما در فایل ./include/nwcommon.h می باشد:

```
/* protocol number for IP */
#define IPPROTO_TCP 6
#define IPPROTO_UDP 17
#define IPPROTO_OSPF 87
#define IPPROTO_BELLMANFORD 520
#define IPPROTO_FISHEYE 530
#define IPPROTO_AODV 123
#define IPPROTO_NEWPROT 123
#define IPPROTO_DSR 135
#define IPPROTO_ODMRP 145
#define IPPROTO_LAR1 110
#define IPPROTO_ZRP 133
```

در این مورد، ما شمارهٔ پروتکل AODV علامت گذاری می کنیم چون در حقیقت پروتکل NEWPROT همان AODV است. در نهایت ما فایل ./application/application.pc را برای نشان دادن شبیه سازی که NEWPROT در سطح شبکه تعریف شده و در سطح کاربرد نمی باشد را تغییر می دهیم.

```
void GLOMO_AppInit(GlomoNode *node, const GlomoNodeInput *nodeInput)
{
. . .
retVal = GLOMO_ReadString(node->nodeAddr, nodeInput,
"ROUTING-PROTOCOL", buf);
if (strcmp(buf, "BELLMANFORD") == 0) {
node->appData.routingProtocol = APP_ROUTING_BELLMANFORD;
```

```
RoutingBellmanfordInit(node);
}
else if (strcmp(buf, "OSPF") == 0) {
// Protocol is at routing layer.
}
else if (strcmp(buf, "WRP") == 0) {
node->appData.routingProtocol = APP_ROUTING_WRP;
RoutingWrpInit(node, nodeInput);
}
. . .
/* Fine because AODV and NEWPROT are defined at network layer*/
else if (strcmp(buf, "AODV") == 0) {
}
else if (strcmp(buf, "NEWPROT") == 0) {
}
. . .
```

هنگامی که این تغییرات کامل شود، دوباره GLOMOSIM را با اجرای فرمانی که در فهرست راهنمای اصلی یجاد شده است، می سازیم. وقتی ما هر شبیه سازی را که NEWPROT را به عنوان پروتکل مسیریابی تعریف می کند، اجرا کنیم، در فایل GLOMO.Stat بررسی خواهیم کرد که پروتکل Aovd استفاده شده است یا نه.

٦- نصب و خطایابی:

٦-١- نصب

پس از آنکه فایل دانلود شدهٔ GLOMOSIM از حالت فشردگی بیرون آمد، دو فهرست فرعی را می توان در آن یافت: glomisim ، parsec ، PARSEC از پیش کامپاسل شده، مجموع برنامه ها را در سیستم عامل های زیر اجرا می کند که در فهرست راهنمای parsec می باشند:

./parsec/aix/	IBM AIX with xlc compiler
./parsec/windowsnt-4.0-vc6/	MS Windows NT or 2000 with Visual C++ ver. 6.0
./parsec/freebsd-3.3/	FreeBSD 3.3
./parsec/redhat-6.0/	Red Hat 6.0 or higher
./parsec/solaris-2.5.1/	Sun SPARC Solaris 2.5.1 or higher with gcc/g++
./parsec/solaris-2.5.1-cc/	Sun SPARC Solaris 2.5.1 or higher with Sun C compiler
./parsec/x86-solaris-2.5.1/	Sun X86 Solaris 2.5.1 or higher with gcc/g++
./parsec/irix-6.4/	SGI IRIX 6.4 or higher with gcc/g++

در سیستم هایی بر مبنای یونیکس، اگر ما دستوری برای ایجاد فهرست /usr/local/parsec داشته باشم، ما فقط کل فهرست های فرعی را با نام سخت افزار پایه، تحت عنوان /usr/local/parsec کپی می کنیم.
اگردستورالعملی برای ایجاد یک فهرست تحت عنوان /usr/local نداشته باشیم، آنگاه در جای دیگری فهرستی را ایجاد می کنیم و فهرست طراحی شده را برای یک محیط متغیر مثل pcc-DIRECTORY تنظیم کنید. برای مثال شما می توانید متغیر را با تایپsetenv PCC- DIRECTORY/home/example/parsec

تنظیم کنید. متغیرهای محیط parsec و گزینه های کامپایلر pcc، متعیرهای محیط را به هنگام اجرا به صورت زیر بررسی می کنند:

PCC_DIRECTORY : Directory that pcc looks up

PCC_CC : C compiler used for preprocessing and compiling

PCC_LINKER : Linker used for linking

PCC_PP_OPTIONS : Options for preprocessing

PCC_CC_OPTIONS : Options for compiling

PCC_LINKER_OPTIONS : Options for linking

در طول نصب، چندین خطا ایجاد می شود که این در صورتی اتفاق می افتد که گزینه ها به درستی نصب نشده و یا فهرست راهنمای parsec /<platform > /bin/ در متغیر PATH نباشد.

GLOMOSIM نیاز به کامپلر C دارد تا با کامپلر ++C/ برای بسیاری از سخت افزارهای پایه اجرا شود. با این وجود در مورد Linux Red Hat G.0 ، نسخهٔ gcc ۲/۹۶ به طور کامل ANSI استاندارد را پشتیبانی نمی کند. بنابراین گونرینه وجود دارد ک هر کدام با نسخهٔ gcc ۲/۹۵ جایگزین شده و یااز نسخهٔ egcs با تعریف متغیرهای زیر استفاده می شود:

setenv PCC_CC egcs

setenv PCC_LINKER egcs++

٦-۲ ابزار برنامه نویسی

برای نصب ابزار برنامه نویسی (VT)، نسخهٔ jave Developmemtkit (JDK) نصب می شود و تمام فایل های مرجع GLOMOSIM کامپایل شده را می بایست ارائه داد. سپس فایل های Jave VT با اجرای زیر کامپایلمی شود:

> cd $glomo/java_gui/

> javac *.java

خطای خیلی معمولی که در هنگام اجرای VT صورت می گیرد این است که با گزینه های Read Time Write Trace شروع می شود و روی صفحه نشان داده نمی شود. راه حل معمولی برای این مسئله، کپس کردن فایل های ترکیب بندی می باشد. (nodes.input,mobility.in,app canf) به تحت عنوان فهرست java-gui/ انجام می شود.

گزینه دیگر، مشخص کردن مسیر کامل VT که در آن فایل ترکیب بندی (CONFIG.IN) قرار داده می شود. با این وجود در این مورد، مسیر تمام فایل های ترکیب بندی فرعی مانندجابه جا پذیری، گره و ترکیب بندی کاربردی را می بایست در فایل CONFIG.IN مشخص کرد.

مسئله دیگری که باید در ذهن داشت این است که وقتی با VT کار می کنید، ارائه دامنهٔ انتقال واقعی با پارامترهای ترکیب بند می تون تعریف می شود. این یکخطای احتمالی است که در تولیدات بعدی تصحیح می شود.

References

[1] GloMoSim: Global Mobile Information Systems Simulation Library. http://pcl.cs.ucla.edu/

projects/glomosim/.

[2] Mario Gerla Lokesh Bajaj, Mineo Takai, Rajat Ahuja, Rajive Bagrodia. GloMoSim: A Scalable Network

Simulation Environment. Technical Report 990027, University of California, 13, 1999.

[3] Addison Lee - Kaixin Xu. GloMoSim Java Visualization Tool. documentation version 1.1, Software

Distribution.

[4] Rajive Bagrodia. README ¯le - GloMoSim Software. University of California, Los Angeles - De-

partment of Computer Science. Box 951596, 3532 Boelter Hall, Los Angeles-CA 90095-1596 / ra-

jive@cs.ucla.edu.

[5] Theodore S. Rappaport. Wireless Communications: Principles and Practice. Prentice Hall, New Jersey,

1999.