



Department of Electrical Engineering

Data Communication Networks

Prof. Pakravan

Course Project

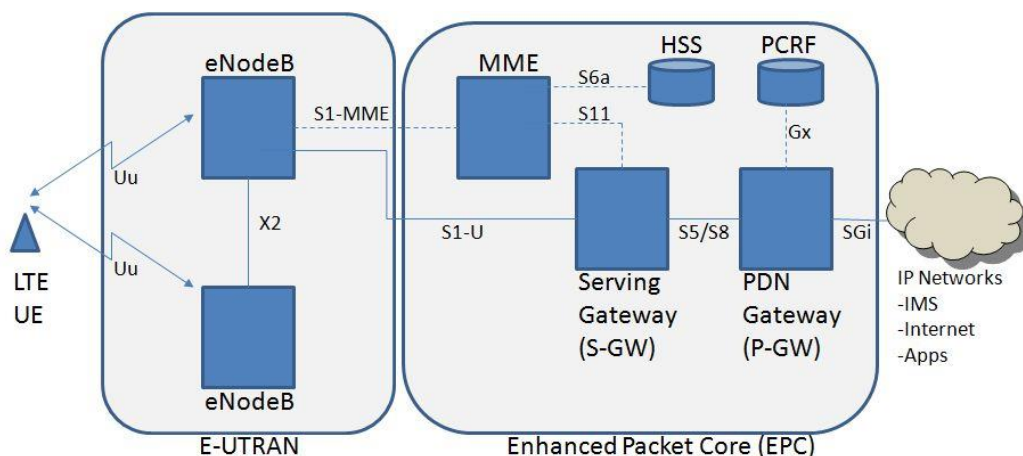
Spring 2020

مقدمه

امروزه شبکه‌های تلفن همراه بسیار گسترش یافته‌اند و استفاده از آن‌ها تنها به انتقال صدا و برقراری تماس محدود نمی‌شود. از همین رو و با افزایش کاربری این شبکه‌ها نیاز به تغییر و بازنگری معماری آنها و بروزرسانی آن همراه احساس می‌شود. در چنین شرایطی استانداردهای متنوعی برای این شبکه‌ها در نسل‌های مختلف تعیین و توسعه داده شده است. از جمله فراگیرترین این استانداردها می‌توان به استاندارد LTE در نسل چهارم شبکه‌های تلفن همراه (4G) اشاره کرد که از اهداف آن می‌توان موارد زیر را نام برد:

- افزایش نرخ انتقال داده
- کاهش تاخیر
- بهینه سازی شبکه برای ارتباطات مبتنی بر بسته
- پشتیبانی از انعطاف در پهنای باند برای ارائه کیفیت سرویس متنوع

این استاندارد که در ادامه سری استانداردهای نسل چهارم ارائه شد معماری ای بصورت زیر دارد که هر یک از اجزای آن کارکرد خاص خود را دارند که در ادامه به بررسی آنها خواهیم پرداخت.



اجزای این استاندارد عبارت هستند از:

- **(UE) User Equipment**
این بخش از شبکه در واقع همان کاربر است یا به عبارت بهتر گوشی تلفن همراهی که قرار است از طریق اینترنتیسی بی‌سیم به شبکه متصل شود.
- **eNodeB**
این بخش از شبکه که در واقع جز اصلی Radio Access Network است، آنتن‌هایی است که در محل‌های مختلف رسم شده و از یک سمت دیگر از طریق اینترنتیسی‌های S1 به بخش Enhanced Packet Core متصل هستند.
- **(S-GW) Serving Gateway**
به عنوان اولین محلی که بسته‌های دریافت شده از طریق آنتن‌های رادیویی وارد شبکه هسته میشوند این بخش اهمیت زیادی دارد از جمله کاربری‌های آن می‌توان به موارد زیر اشاره کرد:

- ارسال داده‌ها به و دریافت آن‌ها از eNodeB و تعیین مسیر داده‌های عبوری
- اولین نقطه مسئول برای مدیریت HandOver بین eNodeB ها
- نگهداری پارامترهایی نظیر حامل‌های IP و یا اطلاعات مربوط به Routing

• (MME) Mobility Management Entity

واحد مدیریت انتقال کاربرها از اصلی‌ترین بخش‌های شبکه‌های تلفن همراه است. در واقع این نود مسئول اصلی مدیریت signaling در شبکه هسته است. بنابر آنچه که در استاندارد آورده شده است مسئولیت‌های اصلی این واحد ایجاد کانال ارتباطی و هویت سنجی کاربرهاست. همچنین اطلاعات مربوط به مکان فیزیکی کاربرها و اینکه بسته‌ها از چه طریقی باید به کاربر ارسال شوند درین بخش نگاه‌داری میشود. همچنین این بخش با جمع‌آوری داده‌های مربوط به محل کاربرها تصمیم میگیرد که کدام eNodeB باید دیتای کاربر را دریافت و به آن ارسال کند.

• (HSS) Home Subscriber Server

این بخش در واقع دیتابیزی است که در آن اطلاعات مربوط به کاربران، شماره آنها و نوع سرویسی که استفاده میکنند در آن ذخیره میشود.

• (PCRF) policy and charging rules function

وظیفه این بخش که در ارتباط با P-GW تعریف میشود، شامل کنترل سیاست‌های کلی شبکه و نیز محاسبه هزینه‌های مربوط به هر کاربر است.

• Packet Data Network gateway

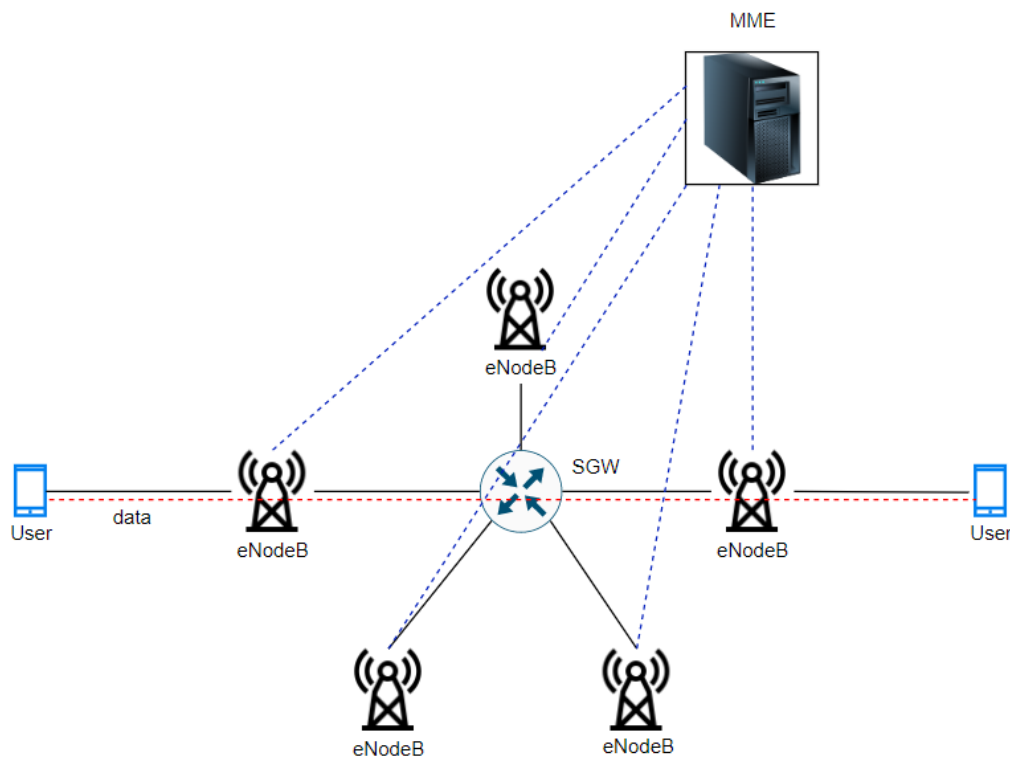
این بخش که اولین محلی‌ست که با شبکه خارج از هسته در ارتباط است به عنوان یک دروازه بین شبکه‌ی خارجی و شبکه تلفن همراه عمل میکند از وظایف اصلی این بخش میتوان به موارد زیر اشاره کرد:

- ارسال بسته‌ها به و دریافت آنها از شبکه مبتنی بر بسته‌ی خارجی.
- محلی که تعیین میکند بسته‌ها به چه نوع شبکه خارجی وارد شوند.
- اعمالی نظیر فیلتر کردن بسته‌ها.
- محاسبه هزینه‌ها در ارتباط با PCRF

پیاده سازی

اجزایی که در این پروژه شبیه سازی خواهند شد به قرار زیر هستند:

- **User**
کاربر متصل به شبکه به دنبال انتقال رشته ای کاراکتر به کاربری دیگر است بدون این که ترتیب آن جابجا شود. هر کاربر با یک ID از دیگری تمییز داده می شود.
- **eNodeB**
اولین نقطه اتصال کاربر به شبکه است. و نیز بایستی به MME اطلاع دهد که چه کاربرانی تحت پوشش او هستند.
- **SGW**
محل اتصال eNodeB های مختلف است. شامل یک جدول مسیریابی است که تعیین می کند دیتای هر کاربر به eNodeB بایستی ارسال شود .
- **MME**
مسئول انجام هماهنگی های لازم جهت جابجایی کاربر از محدوده یک eNodeB به eNodeB دیگر است. این بخش با دریافت موقعیت کاربر از eNodeB مسیر جابجایی دیتا را در SGW تنظیم می کند .



سناریو:

با شروع شبیه سازی از کاربر خواسته خواهد شد که فایلی که در اختیار دارد را برای کاربر مشخص دیگری ارسال کند. توجه شود که این ارسال دیتا باید با رعایت ترتیب باشد، یعنی شبکه بایستی تمهیدات لازم را برای رسیدن دیتا به دست گیرنده با رعایت ترتیب بیندیشد.

این کاربر بر اساس رشته ای از موقعیت ها که در ابتدای کار به شما داده می شود بایستی در محیط شبیه سازی جابجا شود.

مرحله صفرم: در این مرحله براساس ورودی داده شده به برنامه تعداد eNodeB ها و محل قرارگیری آنها و تعداد کاربران و مسیر حرکت آنها و همچنین فاصله زمانی بین هر دو موقعیت کاربر تعیین میشود. پس از دریافت ورودی کد باید موارد زیر را پیاده سازی کند:

- به هر eNodeB باید یک unique Identifier (UID) در ابتدای کار توسط کد داده شود.
- همه eNodeB ها باید با لینکی به سرور sgw متصل شوند. همچنین باید همه eNodeB ها به واحد MME با لینکی متصل شوند.
- واحد MME نیاز است تا با لینکی به sgw متصل شود تا در مراحل بعدی بتواند مسیر ترافیک عبوری را در آن مشخص کند.

مرحله اول: ایجاد کانال سیگنالینگ بین کاربر و شبکه

- هر کاربر باید در زمان اتصال به شبکه یک لینک سیگنالینگ به تمامی eNodeB ها ایجاد کند و در تناوب زمانی مشخص که همان فاصله زمانی داده شده در مرحله صفرم است، موقعیت مکانی خود را به همراه UID مخصوص به خود به تمامی آن ها اعلام می کند.
- eNodeB ها هر کدام مستقلاً به واحد MME متصل هستند و بعد از دریافت موقعیت مکانی کاربر از خود، فاصله خود تا کاربر را محاسبه کرده و آنرا به MME اعلام میکنند تا با تشخیص MME، نزدیکترین eNodeB اعلام شود تا با ارسال پیامی به کاربر به آن متصل شود و آن کاربر را در شبکه ثبت کند و به این ترتیب یک لینک دیتا برای جابجایی داده با آن ایجاد کند.

مرحله دوم: ایجاد نشست انتقال پیام (Session)

- زمانی که کاربر A می خواهد با B انتقال پیام داشته باشد، بایستی با ارسال پیامی به eNodeB خود، درخواست خود را اعلام کند. پس از آن، پیام او از طریق شبکه به دست کاربر B خواهد رسید و نیز کاربر B بایستی آمادگی خود را برای دریافت دیتا به کاربر A با فرستادن پیامی اعلام کند.
- کاربر A با دریافت پیام آمادگی کاربر B، در بسته هایی مشخص، دیتای خود را برای او ارسال می کند. توجه شود که هدف آن است این دیتا با ترتیب ارسالی در گیرنده دریافت شود.

مرحله سوم: مرحله تعویض کانال (Handover)

در سمت فرستنده:

- از آنجایی که کاربر در شبکه در حال حرکت است، ممکن است با تغییر موقعیت مکانی آن فاصله اش از eNodeB که به آن متصل است بیشتر از فاصله تا eNodeB جدید شود. به عبارت دیگر کاربر وارد یک سلول جدید شود، در این حالت با توجه به این که کاربر پس از هر تغییر موقعیت همواره مکان خود را به شبکه اعلام میکند، MME این مساله را تشخیص داده که کاربر در سلول جدیدی قرار گرفته است و این مساله را به eNodeB جدید اعلام میکند.
- پس از آنکه eNodeB جدید از MME پیام مرحله قبل را دریافت کرد به کاربری که وارد سلول مربوط به آن شده اعلام میکند که ارتباط خود را با eNodeB سابق قطع کند و ازین پس داده ها را از طریق لینک دیتای خود برای این eNodeB ارسال کند.
- کاربر پس از دریافت این پیام از eNodeB جدید ارسال داده به eNodeB قبلی متوقف کرده و ادامه آن را به eNodeB جدید ارسال میکند.

در سمت گیرنده:

- اگر بر اثر حرکت گیرنده، سلول مربوط به آن عوض شود نیاز است تا داده ها از طریق مسیر جدیدی ارسال شوند.
- در این حالت نیز چون کاربر در هر قدم مکان خود را به همه eNodeB ها اعلام میکند و آنها فاصله کاربر از خود را به MME اعلام میکنند، پس از آنکه MME متوجه شد که گیرنده وارد سلول جدیدی شده، در ابتدا به eNodeB قبلی اطلاع میدهد که دیگر داده ها را به کاربر ارسال نکند و آنها را بافر کند، همچنین به eNodeB جدید دستور میدهد که به کاربر اطلاع دهد لینک دیتا را با eNodeB سابق قطع کرده و به آن متصل شود.
- پس از اتصال کاربر به eNodeB جدید، eNodeB جدید پیامی به eNodeB سابق میدهد که اگر داده ای مربوط به این کاربر در بافر خود دارد، آن را برای او ارسال کند.
- در این فاصله زمانی ممکن است داده های تازه برای eNodeB جدید از فرستنده ارسال شود که باید آنها را بافر کند.
- برای آنکه ترتیب پیامها حفظ شود باید پس از دریافت محتویات eNodeB قبلی و ارسال آنها به کاربر حال eNodeB جدید میتواند محتویات بافر خود و هر آنچه در ادامه دریافت میکند را برای کاربر ارسال کند.

موارد زیر در شبیه سازی باید در نظر گرفته شود:

- ترتیب داده ها در ارسال و دریافت باید حتما حفظ شود.
- هر کاربر در هر لحظه باید تنها یک کانال دیتا داشته باشد. پس اگر handover روی دهد نیاز است تا کانال دیتای قبلی حذف و کانال دیتای جدید ایجاد شود.
- توجه کنید که شما تنها مجاز به استفاده از قالب های اعلام شده در بخش بعد برای پیامها هستید لذا در هیچ بخشی از شبکه پیامی جز با فرمت آنچه در ادامه اعلام میشود نباید ایجاد و ارسال کنید.

انواع پیام ها

- Position announcement کاربر به eNodeB موقعیت خود را به همراه ID اش را در کانال سیگنالینگ ارسال می کند.
- eNodeB-SGW connection eNodeB باید id خود را به SGW اطلاع دهد.

- **eNodeB-MME connection**
eNodeB باید id خود را به MME اطلاع دهد.
- **Signaling channel setup**
هر کاربر بعد از برقراری لینک، با پیامی ازین نوع id خود را به eNodeB ارسال می کند.
- **My Location**
هر کاربر با این پیام موقعیت خود را به تمامی eNodeB ها از طریق لینک سیگنالیینگ ارسال می کند.
- **User distance**
هر eNodeB با دریافت موقعیت کاربر، فاصله آن کاربر را تا خودش محاسبه کرده و برای MME ارسال می کند.
- **User Registration1**
با مقایسه فاصله کاربر از eNodeB ها، MME بایستی به نزدیکترین eNodeB اعلام کند که کاربر در سلول مربوط به آن قرار گرفته است .
- **User Registration2**
پس از دریافت پیام eNodeB، User Registration1 در کانال سیگنالیینگ به کاربر اعلام می کند که کوتاه ترین فاصله را با او دارد تا کانال جابجایی پیام ایجاد شود.
- **User deregistration**
با ارسال این پیام به eNodeB ای که قبلا مسئول ارسال داده به کاربر بوده اعلام میکنیم که کاربر از سلول آن خارج شده است و ازین به بعد هر داده ای که دریافت میکند و مربوط به آن کاربر است را بافر کند.
- **Create Session**
فرستنده بایستی پیامی به گیرنده ارسال کرده و اعلام کند می خواهد با او نشستگی تشکیل دهد.
- **Create Session Ack**
گیرنده پیام فوق، پس از دریافت آن به فرستنده آمادگی خود را اعلام خواهد کرد.
- **Data Carrier**
این پیام حاوی کاراکتر موردنظر است و از کاربر به کاربر دیگر بر روی کانال دیتا باید منتقل شود.
- **Send Me Buffered data**
پس از اتصال کاربر به eNodeB جدید، eNodeB جدید پیامی به eNodeB سابق میدهد که اگر داده ای مربوط به این کاربر در بافر خود دارد، آن را برای او ارسال کند.
- **Buffered data**
در صورت وقوع handover باید دیتای بافر شده در eNodeB قدیم به eNodeB جدید با این فرمت از پیام ارسال شود.
- **Handover complete**
بعد از ارسال تمامی دیتای بافر شده از eNodeB قدیم به eNodeB جدید بایستی برای eNodeB جدید ارسال شود.
- **Change route**
با تشخیص جابجایی موقعیت کاربر MME به S-GW دستور می دهد مسیر رسیدن دیتا به این کاربر اصلاح شود.

کد بز نیم

سیستمی که شما بر مبنای الگوریتم بالا پیاده سازی می کنید، باید دارای قابلیت های زیر باشد. دقت کند که نوشتن کد نهایی در قالب زیر الزامی است و بخشی از نمره شما را در بر دارد. سیستم نهایی باید به صورت یک کلاس باشد که از شرایط توضیح داده شده در ادامه پیروی می کند و باید توابع زیر را با دقت کامل و با رعایت نحوه ورودی و خروجی شامل شود.

1. Network Constructor

Inputs:

```
#location of eNodeBs  
[(x1,y1) (x2,y2) ....]
```

Outputs:

خروجی این تابع کلاسی می باشد که مجموعه ای از گره ها و مشخصات خواسته شده را در بر دارد. این کلاس تا قبل از تابع بعدی نباید قابلیت ارسال و دریافت پیام بین نود ها را داشته باشد.

Description:

این تابع در ابتدای فراخوان کلاس، اطلاعات لازم در مورد شبکه و توپولوژی آن و همچنین وضعیت نود ها و سرعت لینک ها را دریافت می کند و بر اساس آن ساختار کلی کلاس شکل می گیرد.

```
# Creating a network class  
network = Network( )
```

2. Network Initialization

Inputs:

این تابع نیازی به ورودی ندارد.

Description:

این تابع تمامی سوکت ها را راه اندازی کرده، و ارتباط لازم بین اجزای شبکه شکل می دهد. توجه شود که تا اینجا هیچ یوزری به شبیه سازی اضافه نشده است. پس از پایان این مرحله، راه اندازی اولیه شبکه انجام شده و اجزا طبق توپولوژی مشخص شده در تابع قبلی ارتباط های سوکتی را شکل می دهند و نود ها آماده شروع پروسه ارسال و دریافت پیام هستند.

```
# initialize the network  
network.init_network()
```

3. Add_user

Inputs:

این تابع به عنوان ورودی، یک شناسه دریافت میکند که در ادامه فعالیت خود در شبکه با آن شناخته میشود. به علاوه لیستی از موقعیت ها خواهد گرفت که بایستی در شبیه سازی لحاظ شود.

Outputs:

بدون خروجی

```
# add a user  
network.add_user(id, [(x1, y1), (x2, y2), ...])
```


4. Connection Request

Inputs:

تنها ورودی های این تابع، شناسه گره مورد نظر در شبکه و نام فایل ارسالی می باشد.

Outputs:

بدون خروجی

Description:

این تابع بایستی ابتدا درخواست ایجاد نشست را برای کاربر گیرنده ارسال کند و بعد از دریافت ack، دیتا را از فایل داده شده خط به خط ارسال کند.

```
# Connection Request
```

```
network.connection_request(sender_id, receiver_id, file_name)
```

همچنین برای اینکه با نحوه سوکت پروگرامینگ آشنا شوید یک نمونه کد Client-Server در اختیارتان قرار داده شده است که میتوانید استفاده کنید. همچنین یک چارچوب پیشنهادی نیز برای نحوه پیاده سازی پروژه آورده شده است.

موفق باشید.