به نام خداوند بخشنده‌ی مهربان

الگوریتم‌های مسیریابی در شبکه و شبکه‌های MPLS

استاد حامیم عباسپور
تهیه کننده: علی علی اکبری

به نام آرامش بخش قلب ها

تحقیق و پژوهش فوق ، به منظور تبیین نقش الگوریتم های مسیر یابی در شبکه و همچنین مزایا و معماری شبکه های سلورلار که به GSM موسوم هستند فراهم آمده است.
در این پژوهش از منابع موجود در اینترنت استفاده شده است که در پایان به طور کامل به آنها اشاره کردیم.
همچنین این پژوهش با فرمت PDF در اینترنت قابل دسترسی و دریافت می باشد.

علی علی اکبری
اسفند ۱۳۸۴ و فروردین ۱۳۸۵

فصل اول :

الگوریتم های مسیر یابی در شبکه

اصول عملکرد

روتر ها از الگوریتم های مسیر یابی برای یافتن بهترین مسیر تا مقصد استفاده می نمایند. هنگامی که ما در مورد بهترین مسیر صحبت می کنیم ، پارامتر هایی همانند Hop ها (مسیری که یک بسته از یک روتر دیگر در شبکه منتقل می شود) و زمان تغییر و هزینه ی ارتباطی ارسال بسته را در نظر می گیریم. مبتنی بر اینکه روتر ها چگونه اطلاعاتی در مورد ساختار یک شبکه جمع آوری می نماید و نیز تحلیل آنها از اطلاعات برای تعیین بهتر مسیر.

ما دو الگوریتم مسیر یابی اصلی را در اختیار داریم :

۱) الگوریتم های میسر یابی عمومی
۲) الگوریتم های مسیر یابی غیر متمرکز

در الگوریتم های مسیر یابی غیر متمرکز ، هر روتر در مورد همه ی روتر های موجود در شبکه ، اطلاعات در اختیار ندارد. این الگورتم ها به نام الگوریتم های DV (Distance Vector) معروف هستند. در الگوریتم های مسیر یابی عمومی ، هر روتر اطلاعات کاملی در مورد همه ی روتر های دیگر شبکه و نیز وضعیت ترافیک شبکه در اختیار دارد. این الگوریتم ها به عنوان الگوریتم های LS (Link State) معروف هستند. ما در ادامه ی این تحقیق به بررسی کامل الگوریتم های LS می پردازیم.

الگوریتم های LS

در الگوریتم های LS ، هر روز باید مراحل زیر انجام شود :

۱) *روتر هایی را که به لحاظ فیزیکی به آنها متصل می باشد را شناسایی نموده و هنگامی که شروع به کار می کند ، آدرس های IP آنها را*

بدست آورد. این روتر ابتدا یک بسته ی Hello را روی شبکه ارسال می کند. هر روتری که این بسته را دریافت می کند از طریق یک پیام که دارای آدرس IP خود این روتر می باشد به پیام Hello پاسخ می دهد.

۲) *زمان تاخیر مربوط به روتر های مجاور را اندازه گیری نماید (یا هر پارامتر مهم دیگری از شبکه همانند ترافیک متوسط)*. برای انجام این کار ، روتر ها بسته های Echo را روی شبکه ارسال می کنند. هر روتری که این بسته ها را دریافت می کند با یک بسته ی Echo Reply به آن پاسخ می دهد. با تقسیم زمان مسیر رفت و برگشت به ۲ ، روتر ها می توانند زمان تاخیر را محاسبه کنند (زمان مسیر رفت و برگشت ، سنجشی از تاخیر فعلی روی یک شبکه می باشد). توجه شود این زمان شامل زمان های ارسال و پردازش می باشد.

۳) *اطلاعات خود را در مورد شبکه برای سایر روتر ها منتشر نموده و اطلاعات روتر های دیگر را دریافت کند*. در این مرحله همه ی روتر ها دانش خود را با روتر های دیگر به اشتراک گذاشته و اطلاعات مربوط به شبکه را با یکدیگر مبادله می کنند. با این روش هر روتر می تواند در مورد ساختار و وضعیت شبکه اطلاعات کافی به دست آورد.

۴) *با استفاده از الگوریتم های مناسب بهترین مسیر بین هر دو گره از شبکه را شناسایی کند*. در این مرحله روتر ها بهترین مسیر تا هر گره را انتخاب می کنند. آنها این کار را با استفاده از یک الگوریتم همانند الگوریتم کوتاهترین مسیر (Dijkstra) انجام می دهند.

الگوریتم های Dijkstra

در این الگوریتم ، یک روتر مبتنی بر اطلاعاتی کـه از سـایر روتـر هـا جمـع آوری نموده است ، گرافی از شبکه را ایجاد می نماید. این گراف مکان روتـر های موجود در شبکه و نقاط پیوند آنها را به یکدیگر نشان می دهد. هر پیوند با یک شماره به نام Cost یا Weight مشخص می شود. این شماره تـابعی از زمان تاخیر ، متوسط ترافیک و گاهی اوقات تعداد Hop های بین گره هـا می باشد. برای مثال اگر دو پیوند بین یک گره و مقصد وجود داشـته باشـد ، روتر پیوندی با کمترین Weight را انتخاب می کند.

الگوریتم های Dijkstra دارای مراحل زیر می باشد :

روتــر ، گرافــی از شــبکه را ایجــاد نمــوده و گــره هــای منبــع و مقصــد (برای مثال V1 و V2) را شناسایی می کند. سپس یک مـاتریس بـه نـام ماتریس Adjacency را می سازد. در ایـن مـاتریس یـک مشخصـه مبـین Weight می باشـد. بـرای مثـال [i , j] ، وزن یـک پیونـد بـین Vi و Vj می باشد. در صورتی که هیچ پیوند مستقیمی بـین Vi و Vj وجـود نداشـته باشد این وزن به (Weight) به صورت Infinity در نظر گرفته می شود.

روتر یـک مجموعـه رکـورد وضـعیت را بـرای هـر گـره روی شـبکه ایجـاد می نماید. این رکورد دارای سه فیلد می باشد :

۱) فیلد Predecessor :

اولین فیلدی که گره قبل نشان می دهد.

۲) فیلد Length :

فیلد دوم که جمع وزن های از منبع تا آن گره را نشان می دهد.

۳) فیلد Label :

آخرین فیلد که وضعیت گره را نشان می دهد. هر گره می تواند دارای یک مود وضعیت باشد :

Tentative یا Permanent

روتر ، پارامتر های مجموعه رکورد وضعیت برای همه ی گره ها را آماده سازی اولیه نموده و طول آنها را در حالت Infinity و Label آن را در وضعیت Tentative قرار می دهد.

روتر ، یک گره T را ایجاد می کند. برای مثال اگر V1 می بایست گره T منبع باشد ، روتر برچسب V1 را در وضعیت Permanent قرار می دهد. هنگامی که یک Label به حالت Permanent تغییر می کند دیگر هرگز تغییر نخواهدکرد. یک گره T در واقع یک Agent می باشد.

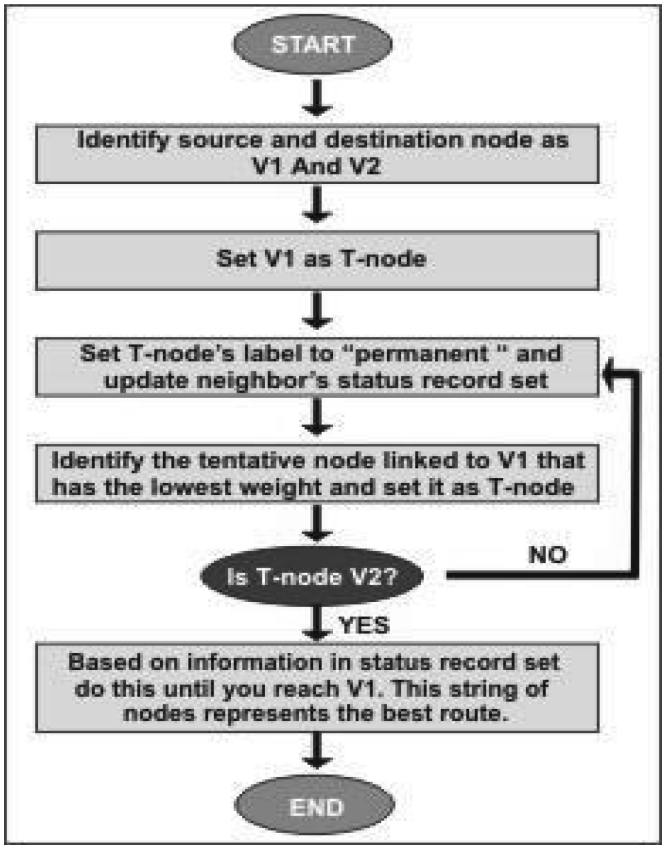
روتر ، مجموع رکورد وضعیت مربوط به همه ی گره های Tentative را که به طور مستقیم به گره T منبع متصل هستند ، روزآمد می نماید.

روتر همه ی گره های Tentative را بررسی نموده و گره هایی را که وزن آن تا V1 کمترین مقدار را دارد ، انتخاب می کند. سپس این گره ، گره T مقصد خواهد بود.

اگر این گره ، V2 نباشد (گره مقصد) ، روتر به مرحله ی ۵ باز می گردد.

اگر این گره V2 باشد ، روتر گره ی قبلی آن را از مجموع رکورد وضعیت استخراج نموده و این کار را انجام می دهد تا به V1 برسد ، این فهرست از گره ها ، بهترین مسیر از V1 تا V2 را نشان می دهد.

این مراحل به صورت یک فلوچارت در شکل نشان داده شده است. اما از این الگوریتم به عنوان یک مثال در ادامه ی تحقیق استفاده خواهیم نمود.

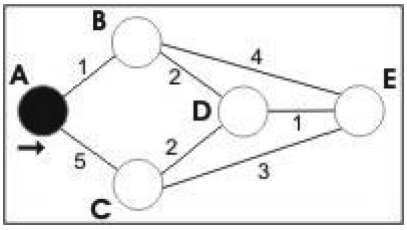


مثال برای الگوریتم Dijkstra

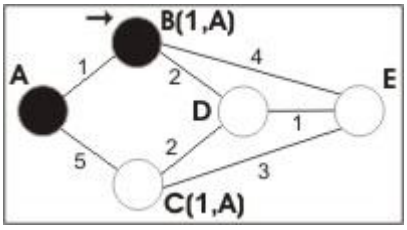
در اینجا ما می خواهیم بهترین مسیر بین گـره هـای A و E را پیـدا کنـیم. همانطور که در شکل صفحه ی بعد می بینید ، ۶ مسـیر بـین A و E وجـود دارد. مسیر هایABE , ACE , ABDE , ACDE , ABDCE , ACDBE و واضح است که ABDE بهترین مسیر مـی باشـد ، زیـرا کمتـرین وزن را دارد. اما همیشه به این سادگی نیست و برخی موارد پیچیده وجود دارد که در آن ما مجبوریم از الگوریتم هایی برای یافتن بهترین مسیر استفاده کنیم.

همانطور که در تصویر ذیل مشاهده می کنید ، گره منبع (A) به عنوان گـره T انتخاب شده است و بنابراین برچسب آن Permanent می باشد.

گره های Permanent را با دایره های تو پر و گره های T را با یک پیکان نشان می دهیم.

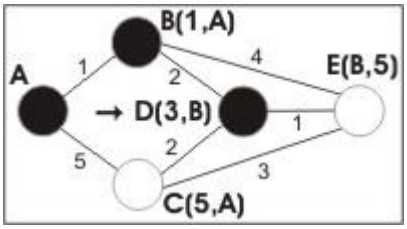


در ایـن مرحلـه شـما مـی بینیـد کـه مجمـوع رکـورد وضـعیت گـره هـای Tentative که به طور مستقیم به گره (C) T ، B متصل شده انـد ، تغییـر یافته است. همچنین از آنجایی که گره B کمتـرین وزن را دارد ، بـه عنـوان گره T انتخاب شده و برچسب آن به حالت Permanent تغییر کرده است.

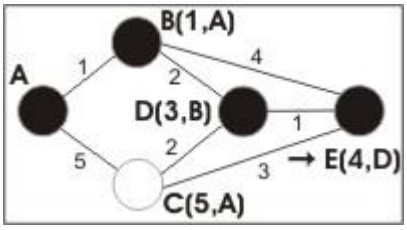


در این مرحله همانند مرحله قبل دو مجموعه رکورد وضعیت گره هـایی کـه Tentative دارای اتصال مستقیم به گره T می باشد (D ، E) تغییر کرده است. همچنین از آنجـایی کـه گـره D وزن کمتـری دارد ، بعنـوان گـره T انتخاب شده و برچسب آن به وضعیت Permanent تغییر کرده است.

در این مرحله ما هیچ گره Tentative نداریم ، بنابراین فقط گره T بعدی را شناسایی می کنیم. از آنجایی که E دارای کمترین وزن می باشد بعنوان گره T انتخاب می شود.



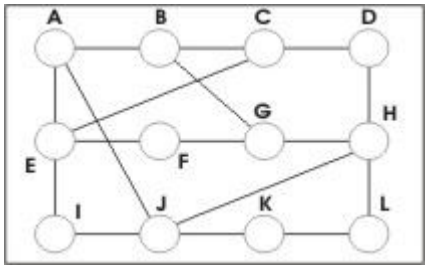
E گره مقصد بوده ، بنابراین کار ما در اینجا تمام مـی شـود. اکنـون مـا کـار شناسایی مسیر را به انتهـا رسـانده ایـم. گـره قبلـی E ، گـره D و گـره B می باشد و گره قبلی B ، گره A می باشد. بنابراین بهتـرین مسـیر ABDE است. در این مورد وزن کل مسیر ، (۱+۲+۱) ۴ می باشد.



با وجودی که این الگوریتم به خوبی کار می کند اما آنقدر پیچیده اسـت کـه زمان پـردازش آن بـرای روتـر طـولانی بـوده و رانـدمان شـبکه را کـاهش می دهد. .همچنین اگر یک روتر اطلاعات غلطی را به روترهای دیگر بدهد ، همه تصمیم های مسیر یابی نادرست خواهد بود.

الگوریتم های DV

الگوریتم های DV با نام های الگوریتم های مسیریابی Bellman-Ford و Ford-Fulkerson نیز یاد می شوند. در این الگوریتم ها ، هـر روتـر دارای یک جدول مسیریابی می باشد کـه بهتـرین مسـیر تـا هـر مقصـد را نشـان می دهد. یک گراف معمولی و جدول مسیریابی مربوط به روتـر G در شـکل زیر نشان داده شده است.



همانطور که در جدول مشاهده می کنید ، اگر روتر G بخواهد بسته هـایی را به روتر D ارسال کند ، می بایست آن ها را به روتر H ارسال نماید. هنگامی که بسته ها به روتر H رسیدند ، این روتر جدول خود را بررسی نموده و روی چگونگی ارسال بسته ها به D تصمیم گیری می کند.

Line	Weight	Destination
A	8	A
A	20	B
I	28	C
H	20	D
I	17	E
I	30	F
H	18	G
H	12	H
I	10	I
-	0	J
K	6	K
K	15	L

در الگوریتم های DV ، هر روتر می بایست مراحل ذیل را انجام دهد :

۱) وزن لینک های مستقیماً متصل به آن را اندازه گرفته و این اطلاعات را در جدول خود ذخیره کند.

۲) در یک دوره زمانی خاص ، روتر جدول خـود را بـه روترهـای مجـاور ارسال نموده و جـدول مسـیریابی هـر یـک از روترهـای مجـاور خـود را دریافت می کند.

۳) مبتنی بر اطلاعات بدست آمده از جداول مسیریابی روترهای مجاور ، جدول خود را روز آمدسازی می نماید.

یکـی از مهمتـرین مشـکلات ، هنگـام کـار بـا الگـوریتم هـای DV ، مشکل Count to Infinity است. اجازه بدهید این مشـکل را بـا ذکـر یک مثال روشن کنیم.

همـانطور کـه در قسـمت ذیـل نشـان داده شـده اسـت یـک شـبکه را در ذهن خود تصور کنید. همانطور کـه در ایـن جـدول مـی بینیـد ، فقـط یـک پیونـد بـین A و سـایر بخـش هـای شـبکه وجـود دارد. در اینجـا شـما می توانید ، این گراف و جدول مسیریابی همه گره ها را مشاهده کنید :

D	C	B	A	
3 , D	2 , B	1 , A	0 , -	A
3 , D	2 , C	0 , -	1 , B	B
1 , C	0 , -	1 , C	2 , B	C
0 , -	1 , D	2 , C	3 , B	D

اکنون تصور کنید که پیوند بین A و B قطع شود. در این هنگام ، B جـدول خـود را تصـحیح مـی کنـد. بعـد از یـک مـدت زمـان خـاص ، روترهـا جدول های خود را مبادله نموده و بنابراین B جدول مسیریابی C را دریافـت می کند. از آنجایی که C نمی داند چه اتفاقی بـرای پیونـد بـین A و B رخ داده است این اطلاعات را حفظ می کند. B این جـدول را دریافـت نمـوده و فکر می کند که یک پیوند جداگانه بین C و A وجود دارد ، بنابراین جـدول خود را تصحیح نموده مقدار Infinity را به ۳ تغییر می دهد. به همین شکل دوباره روترها جدول های خود را مبادله می کنند. هنگـامی کـه C ، جـدول مسیریابی B را دریافت می کند ، مشاهده می نماید کـه B وزن پیونـد خـود تـا A را از ۱بـه ۳ تغییـر داده اسـت ، بنـابراین C ، جـدول خـود را روزآمـد نموده و وزن پیوند خود تا A را بـه ۴ تغییـر مـی دهـد. ایـن پروسـه تکـرار

می شود تا همه گره ها وزن پیوند خود را تـا A در وضـعیت Infinity قـرار دهند. این وضعیت در جدول ذیل نشان داده شده است.

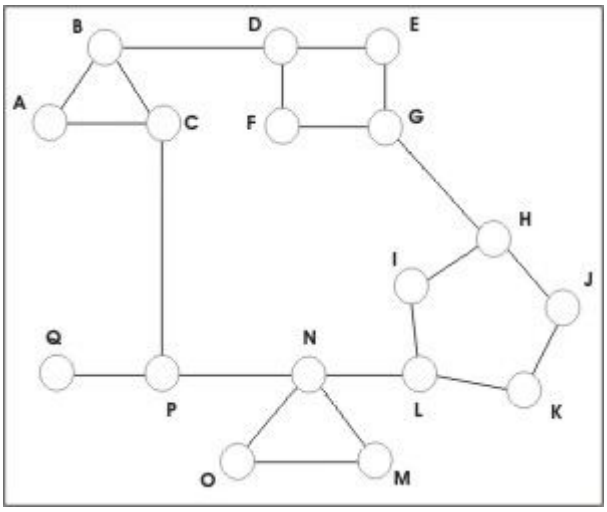
D	C	B	
3,C	2,B	,A	Sum of weight to A after link cut
3,C	2,B	3,C	Sum of weight to B after 1st updating
3,C	4,B	3,C	Sum of weight to A after 2nd updating
5,C	4,B	5,C	Sum of weight to A after 3rd updating
5,C	6,B	5,C	Sum of weight to A after 4th updating
7,C	6,B	7,C	Sum of weight to A after 5th updating

در این روش متخصصین می گویند ، الگوریتم های DV دارای یک سـرعت همگرایی پایین هستند. یک روش برای حل این مشـکل در مـورد روترهـا ، ارسال اطلاعات فقط به روترهایی می باشـد کـه دارای پیونـد انحصـاری تـا مقصد نیستند. برای مثال در این مورد ، C نمی بایست هیچ اطلاعاتی را بـه گره B در مورد A ارسال کند. زیرا B فقط یک مسیر تا A را در اختیار دارد.

مسیریابی سلسله مراتبی

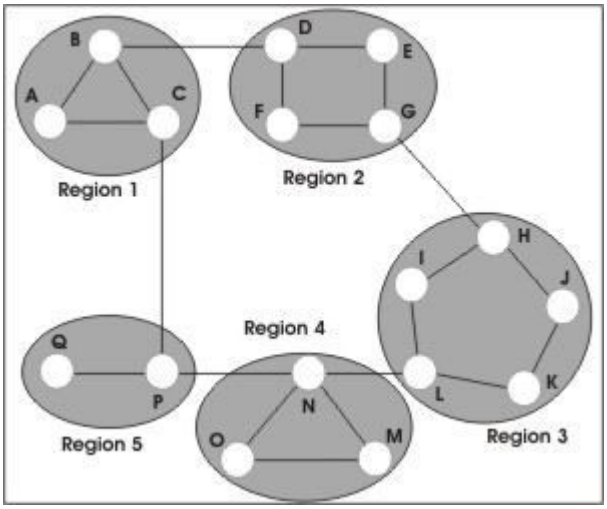
همانطور که شما میبینید ، در هر دو الگوریتم LS و DV ، هر روتـر مجبـور به ذخیره نمودن اطلاعات مربوط به روترهای دیگر می باشـد. هنگـامی کـه اندازه شبکه رشد می کند ، تعداد روترهای شبکه افزایش می یابـد در نتیجـه اندازه جدول های مسیریابی نیز افزایش می یابد و روترها نمی توانند ترافیک شبکه را به طور موثر کنترل کننـد. مـا از مسـیریابی سلسـله مراتبـی بـرای برطرف کردن این مشکل استفاده می کنیم. اجازه بدهید این موضوع با ذکـر یک مثال روشن کنیم :

ما از الگوریتم های DV برای یافتن بهترین مسیر بین گره ها استفاده می کنیم. در وضعیت نشان داده شده در ذیل ، هر گره از شبکه مجبور به نگهداری یک جدول مسیریابی با ۱۷ رکورد می باشد. در اینجا یک گراف معمولی و جدول مسیریابی مربوط به A ارائه شده است.



Weight	Line	Destination
-	-	A
1	B	B
1	C	C
2	B	D
3	B	E
3	B	F
4	B	G
5	B	H
5	C	I
6	C	J
5	C	K
4	C	L
4	C	M
3	C	N
4	C	O
2	C	P
3	C	Q

در مسیریابی سلسله مراتبی ، روترها در گروه هایی به نـام Regions طبقـه بندی می شوند. هر روتر دارای اطلاعـاتی فقـط در مـورد روترهـایی کـه در Region آن ها قرار دارد ، در اختیار داشته و هیچ گونه اطلاعـاتی در مـورد Region های دیگر ندارند. در این مثال ما شبکه خود را بـه پـنج Region تقسیم می کنیم. اگر A بخواهد بسته ها را به هر روتر در Region2 ارسال کند ، آن ها را به B ارسال می کند و الی آخر.



Weight	Line	Destination
-	-	A
1	B	B
1	C	C
2	B	Region 2
2	C	Region 3
3	C	Region 4
4	C	Region 5

در این نوع مسیریابی ، جدول ها را می توان خلاصه نمود بنـابراین رانـدمان شبکه بهبود می یابد. مثال بالا مسیریابی سلسله مراتبی دو سطحی را نشـان می دهد. همچنین میتوان از مسیریابی سلسله مراتبی ۳ سطحی و ۴ سطحی استفاده کرد. در مسـیریابی سلسـله مراتبـی ۳سـطحی ، شـبکه بـه تعـدادی کلاستر تقسیم بندی می شود. هر کلاستر متشکل از تعدادی Region و هر

Region دارای تعدادی روتر می باشد. مسیریابی سلسله مراتبی به طور وسیعی در مسیریابی اینترنت مورد استفاده قرار می گیرد و استفاده از چندین پروتکل مسیریابی را ممکن می سازد.

فصل دوم :

شبکه های GSM

مقدمه

شروع شکل گیری GSM در اروپا ، اوایل دهه ی ۱۹۸۰ است. در آن زمان اروپا با رشد چشمگیر سیستم سلورلار آنالوگ مواجه بود.

C-4500 : آلمان و پرتغال

TACS : اسپانیا ، ایرلند

NMT : اسکاندیناوی

RADICOM : فرانسه

RTMS : ایتالیا

که هیچ کدام با هم سازگار نبودند. به همین علت اتحادیه ی اروپا در سال ۱۹۹۲ یک گروه مطالعاتی تشکیل داد. عنوان این گروه Group Special Mobile بود که بعد ها به Global System For Mobile Communication تغییرنام یافت.

این گروه موظف بود سیستم موبایلی را برای استفاده در سطح اروپا پیشنهاد نماید که جوابگوی موارد زیر باشد :

۱) بهبود کیفیت
۲) سیستم و پایانه ی کم هزینه
۳) پشتیبانی از جا به جایی بین المللی
۴) پشتیبانی از پایانه های دستی
۵) پشتیبانی از سرویس ها
۶) استفاده ی بهینه از طیف فرکانسی
۷) سازگاری با ISDN

امروزه GSM رایج ترین فن آوری نسل دوم است که در ۱۱۰ کشور دنیا راه اندازی شده است.

توانایی های شبکه ی GSM

این شبکه توانایی انتقال دیتا (Data) با سرعت ۹۶۰۰bps را دارد. بـه ایـن معنی که کاربر GSM با این سرعت می تواند به کاربران دیگر در GSM و نیز کاربران شبکه های سوئیچ مداری و سوئیچ پاکتی وصل و بـه مبادلـه ی دیتا بپردازد.

مهمترین سرویس GSM ، همان سرویس تلفنی می باشد ، بـه طـوری کـه ارتباط تلفنی بین کاربران GSM و هر مشترک تلفنی در سراسر دنیـا وجـود داشته باشد.

بخشی از مجموعه استاندارد های GSM برای شـبکه ی تلفـن همـراه مـی باشد. به عنوان مثال شیوه ی عرضه ی سرویس پیام کوتـاه در شـبکه هـای موبایل از استاندارد ۰۳۰۴۰ تعریف شده است.

شبکه ی GSM

GSM یک توصیه کننده می باشد و نوع تجهیزات را مشـخص نمـی کنـد. خصوصیت های GSM اعمال و رابطه ها را مشخص می کند ، ولی سـخت افزار را معرفی نمی کند.

نتیجه ی اخلاقی !!!

۱) کاهش محدودیت طراحان تا حد امکان
۲) افزایش انتخاب در از تهیه کنندگان مختلف

شبکه GSM به سه سیستم اصلی تقسیم می شود :

۱) سیستم سوییچینگ (SS)
۲) سیستم ایستگاه پایه (BSS)
۳) سیستم پشتیبانی و عملیاتی (OSS)

سیستم سوییچینگ (The Switching System)

SS مسئول انجام پردازش های تماس و عملیات مربوطه با مشترک است.

سیستم سوییچینگ از واحد های عملیاتی زیر تشکیل می شود :

۱) ثبت کننده ی موقعیت گذرا (VLR)

۲) مرکز تایید (AUC)

۳) ثبت کننده ی موقعیت دائم (HLR)

۴) ثبت مشخصه ی تجهیزات (EIR)

سیستم ایستگاه پایه (The Base Station System)

تمام اعمال رادیویی در BSS انجام می پذیرد که BSS شامل کنترل کننده های ایستگاه پایه (BSC) و ایستگاه های انتقال پایه (BTS) می باشد.

BSC (Base Station Controllers)

BSC کلیه ی اعمال کنترلی و لینک های فیزیکی بین MTC و BTS را انجام می دهد.

BTS (Base Transceiver Station)

BTS رابط رادیویی به دستگاه موبایل را تنظیم می کند.

سیستم پشتیبانی و عملیاتی (The Operation Support System)

یک نهاد اجرایی است که عملیات شبکه و سیستم های کنترلی را نمایش می دهد. هدف OSS ارائه ی پشتیبانی موثر جهت عملیات مرکزی ، منطقه ای و ناحیه ای و فعالیت های مورد نیاز شبکه ی GSM می باشد.

پایان

استاد حامیم عباسپور

دانشجو علی علی اکبری

با تشکر و قدردانی از :

سایت پویش رایانه

http://POOYESHR.com/

و

مهسا حکمت

برای دریافت نسخه ی **PDF** این تحقیق به وبلاگ

http://EBOOK204.blogfa.com

مراجعه فرمایید.

Ali.Aliakbary@gmail.com

0912 684 21 60