شبکه های پیشرفته

دکتر رامین کریمی

فهرست مطالب

آنالیز مقاله:

قبل از هر چیز باید توجه داشت که مقاله انتخابی دارای ویژگی های زیر باشد :

1 - مقاله Survey **نباشد 2 - مقایسه ای و مروری نباشد**

3 - مقاله ای باشد که پیاده سازی داشته باشد (آخرمقاله یک سری نمودار داشته باشدو یک سری متغییر مقایسه شده باشد)

باتوجه به نکات باید آنالیز نهایی مقاله باید 4 الی 5 اسلاید داشته باشد :

1) فرضیات : که در مقدمه یا Introduction **نوشته شده ، اگر مقاله ای فرضیات نداشته باشد وعنوان آن** Routing protocol In VANET **باشد باید مقاله های** Survey **و موردی در مورد** Keyword **های آن مطالعه گردد**

مقاله Survey **ای خوب است که تقسیم بندی گرافیکی داشته باشد**

مثلا یک مقاله Survey **درمورد** Communication In VANET **میخواهیم که دو نوع دارد**

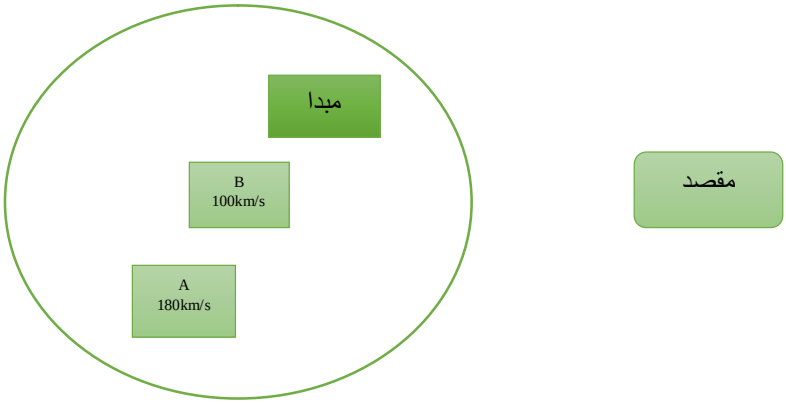
- V2V **: ارتباط ماشین با ماشین**
- V2I **: ارتباط ماشین با ایستگاه کنار جاده و سپس ارتباط ایستگاه با ماشین دیگر**

سپس در فرضیات مثلا می نویسیم که در این مقاله از مدل V2V **استفاده کرده است فرضیات مهم ترین اسلاید است زیرا با حذف یک فرض یا عوض کردن آن میتوان یک مشکل جدید ساخت و یک مقاله جدید نوشت.**

2) Problem : هسته اصلی هر مقاله است . معمولا با شکل نشان می دهند در مقالات مثلا در مقاله ای به نام

Routing Protocol In VANETPrediction Direction **می گوید که یک نود چگونه همسایه هایش را بشناسد ؟ از مکانیزم** Hello Packet **استفاده میشود که هرنود آن را روی بُرد راداری خود می فرستد براساس** ACK **های که دریافت میکند اطلاعات همسایه هایش را بدست می آورد در خیلی از مقالات** Time Duration **مربوط به** Hello Packet **را صفر میگیرند چون ناچیز است (زمان** Hello Packet **یک ثانیه است) ممکن است در یک ثانیه55 متر جابجا شود پس نباید زمان آن را صفر گرفت در شکل زیر مبدا** Hello Packet **را می فرستد و ماشین A ، B هر دو** Detect **میشوند ماشین مبدا بسته را به نودی که به مقصد نزدیک تر است میفرستد پس بسته به A داده میشود و یک ثانیه بعد (باتوجه به سرعت بالای A) در بُرد راداری مبدا نیست ولی مبدا چون محاسبات یک ثانیه قبل را درنظر میگیرد فکر میکند**

نود A در محدوده بُرد راداری اش قرار دارد. پس مبدا بسته را به A می فرستد و چون A نیست بسته Drop میگردد. پس لازم است لحظه بعد پیشگویی شود و براساس ثانیه دوم بسته را بفرستد وبرای B بفرستد .



3) راه حل : راه حل این است که وقتی مبدا باید موقعیت خودش و A را طبق محاسبات ریاضی بررسی کند که آیا A از 1000 متری میگذرد یا خیر که اگر از 1000 متر عبور میکند مبدا بسته را به Bبدهد (در پایان نامه اگر این بخش مشکل داشته باشد FAIL میشود

راه حل به صورت های زیر نمایش داده میشود

- ✓ **فلوچارت** Research Framework
- ✓ **الگوریتم پیشنهادی**
- ✓ Sudo code

4) پیاده سازی یا Implementation

از چه شبیه سازی استفاده شده است ؟ چرا؟ مثلا شبیه سازی NS2 , matlab,opnet,omnet **در مقالات ایرانی از** ns2 , omnet **استفاده میگردد چون** opnet **خیلی گران است و در ایران استفاده نمیشود .**

متغییرها مشخص شود در انتهای مقاله چند نمودار باهم مقایسه شده است که درکنار نمودارها نام متغییرها نوشته میشود

5) نتایج : این اسلاید اختیاری است

6) Future work: اگر مقال future work **داشت آن را مینویسیم اگر کسی مقاله را ادامه ندهد آیندگان میتوانند طبق مواردی که در** future work **ذکر میشود موضوع مقاله را ادامه دهند که حتما باید به نویسنده مقاله ایمیل زده شود.**

مرجع این درس شبکه های پیشرفته نوشته دکتر محمود فتحی عضو هیئت علمی دانشگاه علم و صنعت میباشد

مقدمه

در مقطع کارشناسی با مفهوم Computer Network **آشنا شدیم یعنی شبکه ای که اجزای آن کامپیوتر(**PC**) است ، در مقطع کارشناسی ارشد بحث در رابطه با** Network **است در واقع اجزای شبکه هر چیزی جز** PC **میتوانند باشند .**

تعریف شبکه و شبکه های کامپیوتری متفاوت است ؛ اگر دو Nod **با هم ارتباط و تبادل اطلاعات داشته باشند تشکیل** Network **میدهند بنابر این در این بحث جنس** Nod **ها برای ما اهمیت دارد .**

بنا بر این تقسیم بندی شبکه ها براساس جنس نودها انجام میگردد :

1) Adhoc Network : **اگر دو** Node **ادوات نظامی باشند به آن شبکه** Adhoc Network **گوئیم**

ذات شبکه بحث نظامی بودنش است اولین شبکه که در دنیا به وجود آمد توسط وزارت دفاع در دانشگاه کالیفرنیا به وجود آمد مهد Network **دنیا، دانشگاه کالیفرنیا آمریکا است . تمام** Simulator **ها ، تکنولوژی ها در دانشگاه کالیفرنیا به وجود آمده است دانشگاه کالیفرنیا شبکه** Arpanet **"شبکه جاسوسی " رابرای وزارت دفاع تولید کرد خاصیت تکنولوژی های نظامی این است که تا 2 الی 3 دهه کارکرده و سپس منقضی می شود و فاز تجاری آن آغاز میشود و ازحالت فوق سری خارج شده و تکنولوژی جدید جایگزین می گردد .**

به عنوان مثال در Arpanet **تغییراتی ایجاد کردند و اینترنت به وجود آمد.**

2) Mobile Adhoc Network :

اگر 2 Node **، موبایل باشند شبکه را** Mobile Adhoc Network(MANET) : **گوئیم شبکه** MANET **یک تقریبی از** Adhoc **است.**

تقریب : مثل خورشت قیمه و خورشت قیمه بادمجان در ساختار جدید سخت افزار اضافه میشود اما ساختار اولیه یکی است تمام استراکچر آن تقریب زده میشود .

در MANET **شبکه** Adhoc **وجود دارد ولی برای اتصال هر دو نود نیاز به آنتن است که در اصطلاح به آن** BTS(Base Transcauer station) **می گویند . هر** BTS **دارای بوردی به میزان شعاع 2 کیلومتر است :**

Transmision Range =2 Km

نودها به وسیله آنتن ها به هم وصل می شوند آنتن ها ساختار سلولی دارند که به آن شبکه های سلولی نیز می گویند . مرکز هر سلول یک آنتن است این آنتن ها خطرناک هستند و روی بافت های حساس بدن انسان تاثیر مخربی میگذارند پس حداقل باید 1000 متر دورتر از این آنتن ها زندگی کرد زیرا آنتن ها بدن انسان را شارژ میکنند ، اگر در نزدیکی این آنتن ها زندگی میکنید باید حداقل یک بار در هفته به استخر بروید ، سجده کردن و نماز خواندن به دلیل اینکه در طول روز چندین بار سجده انجام میشود بسیار برای دشارژ شدن موثر است.

شبکه MANET **اولین بار در سال 70 وارد ایران شد درسال 2004 تا 2006 اوج مشکلات این شبکه بود یکی از مشکلات این بود که هنگامی که یک نود ازمحدوده فرکانسی یک** BTS **خارج شده و به محدوده فرکانسی** BTS **دیگر وارد می شد** BTS **اول با** BTS **دوم تماس گرفته و درخواست میکرد یک کانال فرکانسی برای نود خالی کند که عمل** Switching **اتفاق می افتاد که طی این عمل مشکل** hand off **رخ می داد و شبکه قطع می شد و مجبور به برقراری مجدد ارتباط بود.(امروزه این مشکل کاملا حل شده است)**

3)Vehicle Adhoc Network: اگر 2 نود خودرو (Vehicle) باشد شبکه را Vehicular Adhoc Network **گوئیم. که اصطلاحا به این نوع شبکه** VANET **گفته میشود ، استارت و ایده اولیه**VANET **از سال 2003 شروع شده است و تا سه سال دیگر قطعا تمام ماشینها از آن استفاده خواهند کرد .**

یکی از نظریه های معروف استیو جابز در رابطه با این موضوع است که میگویند " برای هر کار ؛ یک اپلیکیشن " این دیدگاه وارد شبکه های VANET **شد ، بنابراین شبکه** VANET **شبکه ای است که براساس** Application **طراحی شده است که به آن** Application In VANET **می گویند .**

برای اتصال دو خودرو به یکدیگر نیازمند تجهیزات صفحه نمایش و یک برد کوچک الکترونیکی یا در اصطلاح (OBU) On Board Unit **هستیم که داخل موتور خودرو وصل می شود ،** OBU **دارای یک حافظه به همراه یک میکرو پروسسور کوچک ، رادار ، یک آنتن با استاندارد** 802.11P **و یک بافر می باشد .**

شبکه های VANET **دارای استاندارد** 802.11p **هستند این به این معناست که ماکزیمم برد راداری یک ماشین (** Max Transmition Range**) در این شبکه 1000 متر است .**

تذکر :

MANET**شبکه های 3 بعدی هستند ولی شبکه** VANET **2بعدی است .برای پیاده سازی شبکه های 3بعدی میتوان از شبیه ساز** NS2 **استفاده کرداما برای پیاده سازی** VANET **علاوه بر شبیه ساز** NS2 **نیاز به شبیه سازیهای دیگری هم می باشد مثلا چون نود درحال حرکت است باید روی یک** map **حرکت کند که برای پیاده سازی آن از شبیه ساز** Sumo **استفاده می گردد هم چنین برای پیاده سازی شتاب و سبقت ماشین باید از شبیه ساز** move **استفاده شود در سه بعدی یک نود هرجا بخواهد میتواند برود اما در شبکه های دو بعدی باید ماشین در مسیر حرکت کند ؛**

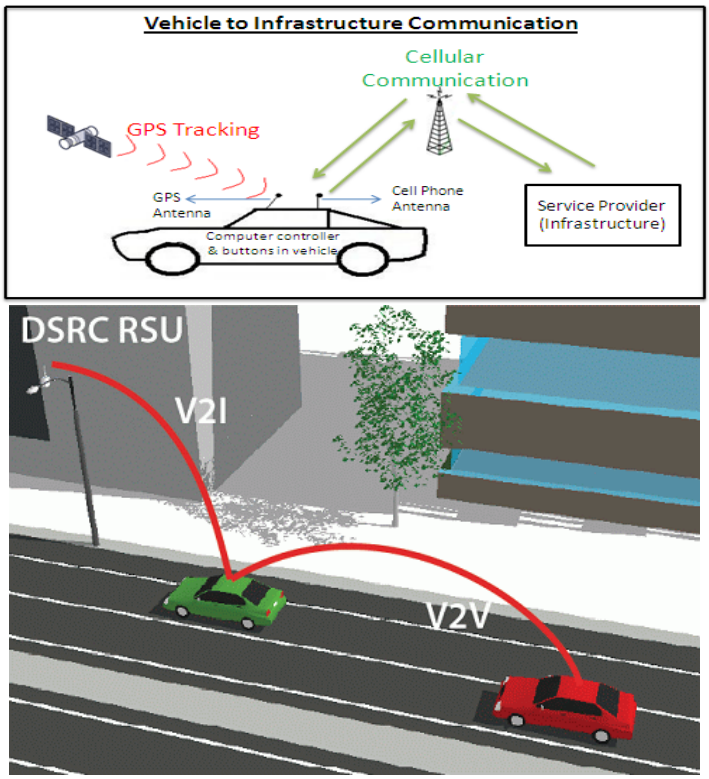
پس برای پیاده سازی پروژه *VANET* ***نیاز به 3 تا*** *Simulator* ***زیر می باشد***

1. *NS2*
2. *Sumo* ***توسط این*** *simulator* ***باید نقشه را تعیین کنیم***
3. *Move*

طریقه ارتباط دو ماشین به دو صورت است :

1) V-to-V (Vehicle to Vehicle) :

بیشتر در Highway **استفاده میشود وبه این صورت عمل میکند که وقتی دو ماشین در برد راداری هم قرار بگیرند با هم کانکت میشوند و از همدیگر مطلع میشوند**(Dedicated short-range communications) **و موقعیت خودشان را از**GPS **دریافت میکنند .**



2) V-to-I (Vehicle to Infrastructure Communication) :

اجزاء اصلی این ارتباطات شامل واحد نرم افزاری (AU) **، واحد آن برد** (OBU) **و واحد تجهیزات کنار جاده (** RSU)**می باشند. در این روش برای ارتباط بین دو ماشین یکی از خودروها به آنتن کنار جاده ای**(Infrastructure) **متصل میشود و آنتن کنار جاده ای به ماشین دوم متصل میشود.**

این مدل به دلیل گران بودن تجهیزات معمولا به صورت شهری و تنها در چهار راه های بسیار شلوغ استفاده میشود .

همانطور که گفته شد شبکه VANET **شبکه ای است که براساس** Application **طراحی شده است؛**

خودروها به سرعت و در کسری از ثانیه می‌توانند از یک حادثه در چند صد متری خود مطلع شده و تغییر مسیر دهند، ترافیک را تشخیص داده و از خودرو جلویی یا عقبی سوال و جواب کنند یا از وضعیت ترافیک خیابان یا کوچه‌های کناری و چهارراه پیش رو اطلاع یابند. شکل می‌تواند به خوبی این وضعیت را نشان دهد. وقتی یک اتفاق ناگهانی در خیابان یا جاده رخ می‌دهد، خودروهای جلویی یا عقبی به سرعت می‌توانند به یکدیگر خبر داده و همچنین با ارتباط‌گیری با ایستگاه‌های مرکزی به پلیس یا مدیریت ترافیک شهری خبر دهند. راننده‌ها با اطلاعاتی که از خودروهای اطراف به دست می‌آورند، می‌توانند تصمیم‌گیری‌های به مراتب مطمئن‌تر و بهتری داشته باشند و یک رانندگی ایمن و راحت‌تر و همراه با لذت را تجربه کنند. کاربردهای شبکه‌های VANET **در شرایط بد آب‌وهوایی نمود بیشتری می‌یابد. در هوای ابری یا مه‌آلود خودروها با ارتباطات خود می‌توانند یکدیگر را راهنمایی کرده و مانع بروز حادثه شوند. در این شبکه‌ها از برنامه امنیتی** SVA (Slow/Stop Vehicle Advisor) **برای هشدار دادن، کم کردن سرعت یا توقف خودرو،** EEBL (Emergency Electronic Brake-Light) **برای ترمز ناگهانی،** RHCN (Road Hazard Control Notification) **برای هشدارهای مرکز کنترل ترافیک،** PCN (Post Crash Notification) **برای ارسال هشدارهای تصادف و** CCW (Cooperative Collision Warning) **برای اعلام هشدار برخوردها استفاده می‌شود.**

سه دسته Application در VANET وجود دارد :

Safety Application - Commercial Application - Entertainment Application

1) Safety Application :

این اپلیکیشن موجب به حداقل رساندن خسارت و صدمات تصادفات رانندگی میشود و هنگامی به اجرا در می آیند که fault **رخ دهد.اما تمام اپلیکیشنهای** Safety **ای که طراحی میگردد برد 300 - 500 متر را درنظر میگیرد.چون در حالت** safety **پیام حتما باید به نود موردنظر برسداین بازه را درنظر میگیرند خارج از این بازه تضعیف سیگنال وجود دارد .**

بررسی چند نوع اپلیکیشن Safety :

1- EEBL : Emergency Electronic Brake Lights

(هنگامی که سه خودرو پشت سر هم حرکت میکنند و خودرو اول ناگهانی ترمز میکند)

2 - LCW :Lane Change Warning **(هشدار تغییر باند حرکت)**

فرض کنید ماشینی درحال حرکت است و ماشین عقبی میخواهد از آن سبقت بگیرد ولی ماشین جلویی آن را نمی بیند

و میخواهد لاین حرکت خود را تغییر دهد پس سیستم داخل ماشین هشدار میدهد: No change laining

3 – BSW : Blind Spot Warning **(هشدار نقطه کور)**

4 – FCW : Forward Change Warning **(هشدار تغییر رعایت فاصله با خودرو جلویی)**

5 – DNPW : Do Not Pass Warning **(هشدار موقع سبقت نسبت با اینکه از روبه رو ماشین می آید)**

6 – IMA : Inter Section Movement Assist **(راهنمایی در تقاطع به هر ماشین یک دستور متفاوت میدهد)**

7 – LTA : Left Turn Assist **(راهنمایی در گردش به چپ)**

Behavior driver : Application **هایی هستند که براساس رفتار راننده طراحی شده اند مثلا ماشینی باسرعت** 100K**درحرکت است ماشین دیگر با سرعت 180 کیلومتر میخواد از آن سبقت بگیرد اما حتما به آن برخورد میکند پس این برنامه قبل از تصادف و طبق بررسی هایی ،** Air bag **راباز میکند تا وقتی ماشین عقبی به آن برخورد کرد سرنشینها آسیب کمتری ببینند.**

Emergency **در حال حاضر پیشرفته ترین اپلیکیشن است که وقتی آن را فعال کنید هم خطوط سفید را میشناسد و کنترل ماشین را در دست میگیرد و همچنین اولین مرکز درمانی توسط** GPS **شناسایی میکند و اعلام میکند به مرکز تا زمانی که به مقصد برسد کنترل ماشین به دست میگیرد .**

2) Commercial Application : این برنامه تمام تراکنشهایی مالی رابه جزء دریافت وجه انجام میدهد که درماشین میتوان این کار را کرد .

3) Entertainment Application : مثلا در یک جاده هستیم و میخواهیم موزیکی را play **کنیم ولی نداریم به ماشین دیگری در آن جاده درخواست میدهیم و موزیک را میفرستد دانلود میکنیم وگوش میکنیم یا با استفاده از این** App **میتوانیم نزدیک ترین پمپ بنزین ، رستوران ، هتل ، مکان های دیدنی و .... را پیدا کرد.درحال حاضر کشورهای دیگر با استفاده از** GPS **از این** App **استفاده میکنند .**

تا اینجا سه نوع نوع شبکه بر اساس جنس Nod **را بررسی کردیم ؛**

4 - Fly Adhoc Network : اگر 2 نود هواپیما یا پهباد باشند شبکه را FANET **میگویند این شبکه از سال 2003 و بعداز** vanet **آمده است این شبکه موضوع جدیدی است و جز شبکه های3 بعدی است.**

موضوع خارج از درس: در ارشد باید موضوعی که وجود دارد را انتخاب مطالعه و بررسی نمود سپس مطلب جدیدی را از آن خارج کرده و چیز جدیدی را بدست آورد . enhancement

الگوریتم Routing **ای در** Adhoc **است به نام** GPSR **که در** MANET ,VANET **نیز استفاده میگردد که میتوان آن را تقریب زد و در** FANET **نیاز استفاده نمود. این الگوریتم دارای مکانیزم** Greedy **است پس در هر شبکه ای به این مکانیزم نیاز است. یکی از** App **هایی که در** FANET **اهمیت دارد** safety Application **است مثلا اگر دو هواپیما وارد برد راداری یکدیگر شوند** Safety App **حالت هواپیما را از** Pilot **به چپ و راست نمیتواند برود و** Auto pilot **یکی از هواپیماها را بالا میبرد و دیگری را پایین می برد تا همدیگر را رد کنند.**

5 - ship Adhoc Network : اگر 2 نود کشتی باشد شبکه را SANET **گوییم.**

6 - Sensor Adhoc Network : اگر دو نودحسگر باشد شبکه را sensor network **گوییم یکی از مهم ترین مباحث شبکه** Sensor Body Network **است که درمورد سنسورهای است که در بدن انسان جایگذاری میگردد مثلا برای اندازه گیری فشار خون انسان هرلحظه استفاده می گردد.**

فصل یک : مقدمه ای بر شبکه 1

دلایل استفاده از شبکه :

1. **استفاده مشترک از منابع**
2. **کاهش هزینه** (Cost model) **پیاده سازی شبکه مدل هزینه باید مشخص باشد هرچه هزینه کم تر باشد مدل بهتر است، به عنوان مثال بجای خرید 40 پرینتر برای هر بخش یک شرکت با یک پرینتر میتوان هزینه را به 40/1 کاهش داد.**
3. **قابلیت اطمینان** (Reliability) **پارامترها باید مشخص باشد مثل** Delay packet delivery ratio **برای** reliability **و** performance **نیز باید متغییرها تعریف گردند**
4. **کاهش زمان** delay **شبکه بهتر است که تاخیر کم تری داشته باشد**
5. **قابلیت توسعه: فرض کنید یک** routing **طراحی شده است که برای مدل** V2V **است و مختص اتوبان است و** routing **دیگری برای** V2I **شهری طراحی شده است اما میتوان** Routing **ای را طراحی کرد که هم برای** V2V **و هم برای** V2I **کارکند و توسعه پذیر باشد** Scaleability
6. **ارتباطات: یکی از بحث های بسیار کاربردی در موضوع ارتباطات بحث شبکه های** LMS **میباشد.** LMS:Learning Management System **که در رابطه با آموزش مجازی است.**

   LMS **به دو بخش تقسیم میشود ؛ آنلاین و آفلاین**

   دانشگاه های مجازی بر اساس LMS **آفلاین کار میکنند به این صورت که دانشجو با مراجعه به سایت درس مورد نظر را بدون حضور استاد مشاهده میکند مراجعه دانشجو میتوانند هر زمان و هر روزی تا قبل از جلسه بعدی باشد در این روش صدا و اسلاید وجود دارد ولی تصویر استاد وجود ندارد.**

   اما در LMS **آنلاین همه دانشجویان به همراه استاد بایستی در یک زمان معین حضور داشته باشند پیاده سازی این سیستم بسیار دشوار تر است و دانش آن بر مبنای** Distributed Systems **یا سیستم های توزیع شده می باشد که شامل سه لایه است :**

   لایه اول Application **/ لایه دوم** Application Server **/ لایه سوم** Database Access

در طراحی شبکه مواردی که قبل از راه اندازی باید درنظر گرفت موارد زیر می باشد :

1) اندازه سازمان : در قدم اول باید اندازه سازمان مشخص شود تاتعداد کلاینت ها و دفاتر وشعب سازمان و... بدست آید براساس آن LAN ,MAN **... مشخص میگردد .**

2) سطح امنیت : که براساس اهمیت شبکه تعیین می گردد و تعیین میکند پارامترهای امنیت اطلاعات چیست .

مثلا در بحث smart house**ها هر دوربین که در خانه تعبیه شده است داری** IP **است یکی از عیب های** Smart house **امنیت آن است زیرا** Smart house **در خارج از ایران نیاز به امنیت بالایی ندارد .کنترل تمام** IP **ها روی** Access point **است ذات** IP **این است که** Scure **نیست در این روش روی تکنیک های امنیت کار نمیکنند روی فایروال آن کنترل دوربین ها نباید روی** IP **یا** ADSL **باشد باید روی شبکه** GPRS **بیاید پس سیم کارت روی بُرد گذاشته و کدتعریف میکنیم و بعد با اس ام اس کد را به سیم کارت فرستاده و** Application **را** Run **میکنیم و دوربین به کار می افتد در** GPRS **نمی توان فایروال را دور زد .**

3)نوع فعالیت : مثلا شبکه برای فعالیتهای آموزشی دارای متغییری است که Data Type **آن** int **است و رنج آن** [- 32768,32767] **2بایتی است و بین اعداد 0 تا20 یا 0 تا100 است هرچه دیتا بزرگ تر باشد ترافیک شبکه تغییر کرده و ممکن است سخت افزار شبکه نیز تغییر کند پس نوع فعالیت با ترافیک شبکه رابطه مستقیم دارد .**

4) مقدار ترافیک : (فصل سوم)

5) سطح مدیریت : که براساس اهمیت شبکه تعیین می گردد و مشخص میکند هرکس به کدام بخش از شبکه دسترسی داشته باشد .

6) بودجه: بودجه سازمان ها و شرکتها باید درنظر گرفته شود .

مدل های شبکه

در یک شبکه یک کامپیوتر می تواند هم سرویس دهنده و هم سرویس گیرنده باشد. یک سرویس دهنده کامپیوتری است که فایل های اشتراکی و همچنین سیستم عامل شبکه که مدیریت عملیات شبکه را برعهده دارد را نگهداری می کند برای آن که سرویس گیرنده بتواند به سرویس دهنده دسترسی پیدا کند ابتدا سرویس گیرنده باید اطلاعات مورد نیازش را از سرویس دهنده تقاضا کند سپس سرویس دهنده اطلاعات درخواست شده را به سرویس گیرنده ارسال خواهد کرد.

سه مدل شبکه ای که مورد استفاده قرار می گیرند به قرار زیراست :

- ✓ Peer To Peer **یا نظیر به نظیر**
- ✓ **مبتنی بر سرویس دهنده** Server based
- ✓ **مبتنی بر سرویس دهنده – سرویس گیرنده** Client-Server

Peer To Peer :

دراین شبکه ایستگاه ویژه ای جهت نگهداری فایل های اشتراکی ، سیستم عامل شبکه وجود ندارد هر ایستگاه میتواند به منابع سایر ایستگاه ها در شبکه دسترسی پیدا کند هر ایستگاه خاص میتواند هم به عنوان سرویس دهنده وهم به عنوان سرویس گیرنده عمل کند . در این مدل هر کاربر خود مسئولیت مدیریت و ارتقاء دادن نرم افزارهای ایستگاه خود را به عهده دارد از آنجای که یک ایستگاه مرکزی برای مدیریت عملیات شبکه وجود ندارد این مدل برای شبکه ای با کمتر از 10 ایستگاه به کار می آید. ایستگاهها به هم وابسته نیستند و مستقل اند .دراینجا همان بحث شبکه BUS **است و تا 10 متر افت سیگنال نداریم .**

نکته) در شبکه Server based **سرورها میتوانند** ..... Application ,Database ,Proxy,FTP **... باشند در سرور** Application **یک مدل 2 لایه ای و مدل 3 لایه ای وجود دارد که مدل دو لایه ای و سه لایه ای معماری شبکه را تعیین میکند در مدل سه لایه ای** Application server **در لایه 2 و** Database server **لایه 3 قرار دارد .**

Server based :

در این مدل شبکه یککامپیوتر به عنوان سرویس دهنده کلیه فایلها و نرم افزارهای اشتراکی نظیر واژه پردازها ، کامپایلرها، بانکهای اطلاعاتی ، و سیستم عامل شبکه را درخودنگهداری میکند یک کاربر می تواند به سرویس دهنده ، دسترسی پیدا کرده و فایلهای اشتراکی را از روی آن به ایستگاه خود منتقل کند مثلا در یک کارخانه Application server **را در کارخانه و** Data base server **در دفتر دبی نگهدای میکنند.**

تفاوت Application server **با** Data base Server **وقتی میگوییم لایه 2 یعنی** Application server **و لایه 3 یعنی** Data base Server **مثلا برای طراحی سایت براساس لایه ها قیمت گذاری داریم مدل شبکه ای برای این نرم افزار طراحی کرده ایم بعضی وقتها** Proxy Server **هم داریم یعنی نرم افزار براساس مدل شبکه ای طراحی میشود (این موضوع در فصل 2 میخوانیم).**

Client / Server :

در این مدل یک ایستگاه درخواست انجام کارش را به سرویس دهنده ارائه می دهد وسرویس دهنده پس از اجرای وظیفه محوله نتایج حاصل را به ایستگاه درخواست کننده برگشت میدهد دراین مدل حجم اطلاعات مبادله شده شبکه در مقایسه بامدل مبتنی بر سرویس دهنده کم تر است و این مدل دارای کارایی بالاتری می باشد مثلا یک PC **درخواست خود را به سرور میفرستد و تا زمان دریافت پاسخ از سرور درحالت** Standby **می ماند اصطلاحا می گویند تا** Transaction **انجام گردد که براساس مدل 2لایه و 3 لایه ،** process **ای که سمت سرور انجام می گردد متفاوت است که در مدل 2 لایه که به آن** thinfat **میگویند می توان سنگینی کار را در دوطرف** Client **و** server **تنظیم کرد.**

در این مدل یک ایستگاه درخواست کارش را به سرویس دهنده ارائه میکند و سرویس دهنده پس از اجرای وظیفه محوله ، نتایج حاصل را به ایستگاه درخواست کننده عودت میدهد . این مدل حجم اطلاعات مبادله شده شبکه در مقایسه با مدل مبتنی بر سرویس دهنده کمتر است و این مدل دارای کارایی بالاتری است.

شبکه های فعال(Active Network**) : بسته ها میتوانند برنامه هایی باشند که در مقصد اجرا میشوند.**

غیر فعال (Passive Network**) : داده ها فقط بسته ها هستند .**

شبکه سلولی : عبارتست از بسته هایی با طول ثابت اطلاعات.

انواع شبکه از نظر جغرافیایی :

1. ***LAN*** **:درحد یک سازمان است مثلا آی پی های داخلی دانشگاه که تحت** IP **اصلی دانشگاه به هم وصل می شوند** LAN**هستند.**

مانند تماس تلفنی بین دو شماره در محدوده یک کشور و هر دو در یک شهر :

+9821720000

+9821324220

2. ***MAN*** **: شبکه شهری است مثل ارتباط شماره های تلفن در یک کشور از دو شهر مختلف :**

+9821720000

+9811728888

گاهی دو LAN **به هم از دو شهر مختلف متصل میشوند که به این شبکه نیز** MAN **میگویند:**

درشکل فوق دو شبکه LAN **با** IP **های مختلف هستند که با هم تحت** IP **یک کشور مرتبط شوند شبکه** MAN **را میسازند در غیر اینصورت شبکه** WAN **میسازند**

3. ***WAN*** **: شبکه جهانی است مثلا اگر شماره تلفنی در انگلیس با کد 44+ با تلفنی در ایران باکد 98+ ارتباط برقرارکند شبکه** WAN**میسازد:**

+98210000

+44888888

4. ***PAN (Personal Area Network)*** **: در** Smarthouse **استفاده می شود که شرکت** ITS **ارائه میدهد ، مثلا در یک خانه هوشمند 10 دیوایس داریم بنابر این به ده عدد** IP **نیاز داریم پس شبکه ای خواهیم داشت با 10 عدد** IP

*** تذکر : هسته اصلی همه شبکه های فوق** IP **است .**

توپولوژی شبکه

تعریف توپولوژی : به طریقه چیدمان نودها در شبکه گفته می شود.

انواع توپولوژی :

- ***STAR :***

در این توپولوژی یک نود درمرکز قرار دارد که می تواند Hub **یا سوئیچ باشد ؛ تفاوت این دو در این اینست که در** Hub **مکانیزم** Broad cast **است و در سوئیچ مکانیزم** Uni cast **است در برخی از کتب عنوان شده است که مکانیزم** Hubچون Broad cast **میباشد بنابراین تولید ترافیک مینماید و این استدلال بیشتر در تئوری درست است اما در عمل چون در** Hub **تعداد نود ها پایین است در نتیجه ترافیک معنایی ندارد .**

ویژگی ها :

در این توپولوژی هر نود به نود دیگر به صورت point to point **ارتباط دارد .**

در Hub **هر دیتایی ارسال شود تمام نودها دریافت می کنند .**

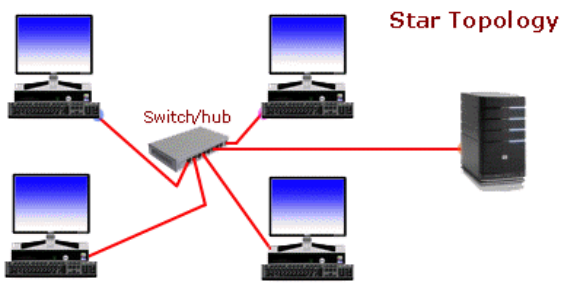
در Hub **،** collision **وجود دارد اگر نود مرکزی از کار بیفتد کل سیستم قطع می شود.**

اگر 5 تا 10 کامپیوتر (تعدادکم) به هم وصل شود از Hub **استفاده می کنیم در غیر این صورت** Switch **استفاده می گردد.**

ترافیک شبکه زمانی مطرح است که تعداد نودها خیلی زیاد باشد ممکن است پهنای باند کم باشد و ترافیک نباشد یا پهنای باند زیاد باشد و ترافیک وجود داشته باشد ترافیک به ورودی یا تروپوت سیستم بستگی دارد وقتی 5 نود در شبکه باشد ترافیک وجود نخواهد داشت .

اگر شبکه کوچک باشد و collision **رخ دهد در میکروثانیه مجددا بسته ارسال می گردد.** Collision **در** Hub **اتفاق نمی افتد چون تعداد نودها کم است .**

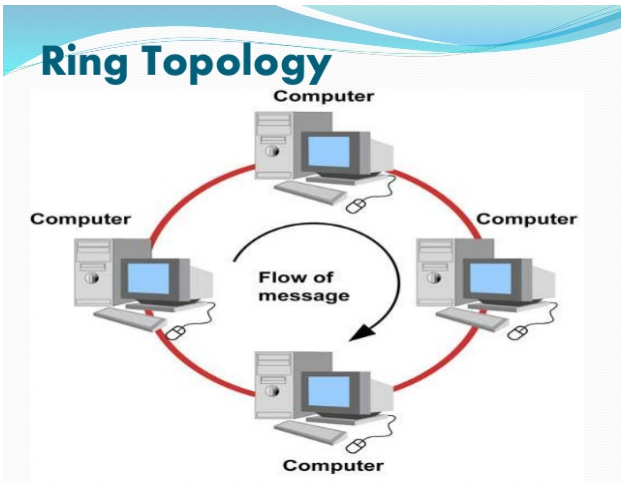
اگر pc3 **بخواهد به** pc4 **داده بفرستد اول به نودمرکزی می دهد و آن هم به همه** Broad cast **میکند** pc4 **بسته را بر می دارد و بقیه آنرا** drop **میکنند اما در** Switch **بسته فقط به** pc4 **ارسال می گردد.**



- ***RING***

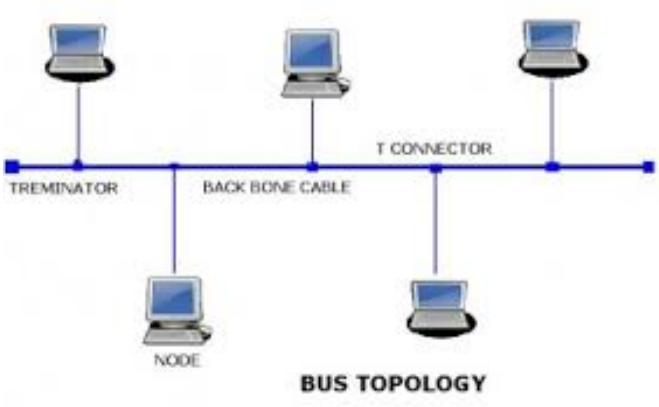
کاربرد ring **در تکنولوژی فیبر نوری است در ایران تکنولوژی فیبر نوری و مخابرات را داریم اما چون زیر ساخت مناسب معماری و شهرسازی نداریم نمی توانیم اینترنت پرسرعت داشته باشیم .**

در این توپولوژی کلیه کامپیوتر ها به گونه ای به یکدیگر متصل میشوند که مجموعه ی آنها تشکیلیک حلقه را میدهد ، کامپیوتر مبدا اطلاعات را به کامپیوتر بعدی در حلقه ارسال میکندو به همین ترتیب این روند ادامه پیدا میکند تا اطلاعات به کامپیوتر مبدا برسد سپس کامپیوتر مبدا این اطلاعات را از روی حلقه حذف میکند.



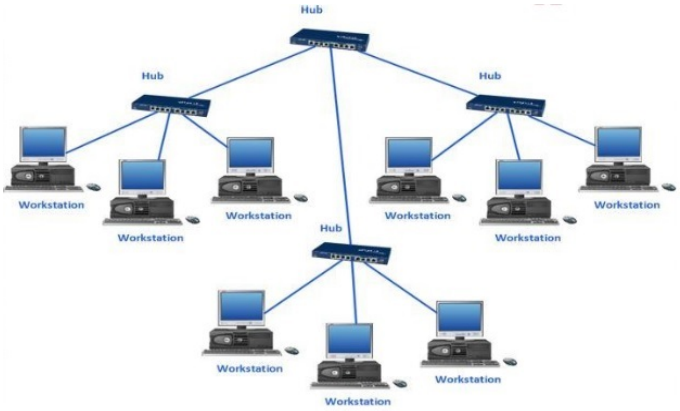
نکته : انواع توپولوژی ها در تمام شبکه های VANET t ADHOC **و ... وجود دارد .**

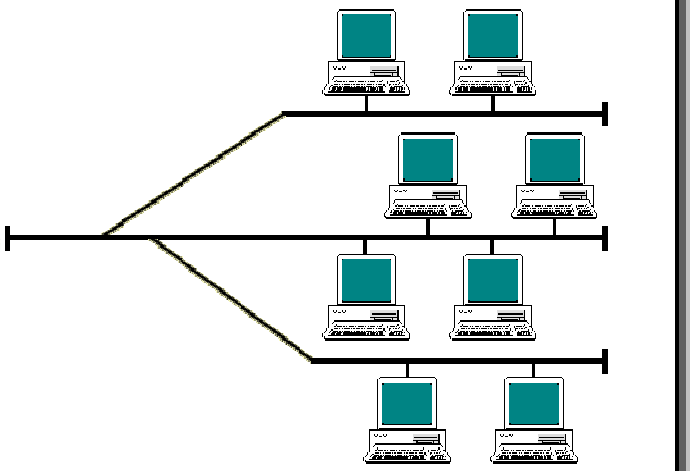
- ***BUS***



• TREE

به عنوان مثال در یک ساختمان چند طبقه تمام کامپیوتر های طبقه اول را با یک سوئیچ و با هر نوع توپولوژی شبکه میکنیم و به همین ترتیب بقیه طبقه ها...حال برای کانکت کردن همه طبقات با هم از یک سوئچ رده بالاتر استفاده میکنیم .

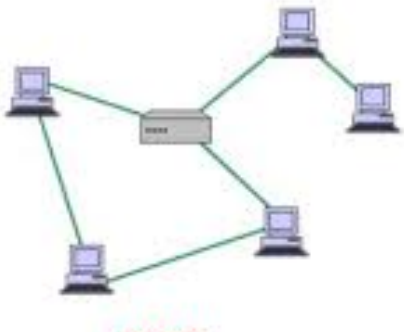


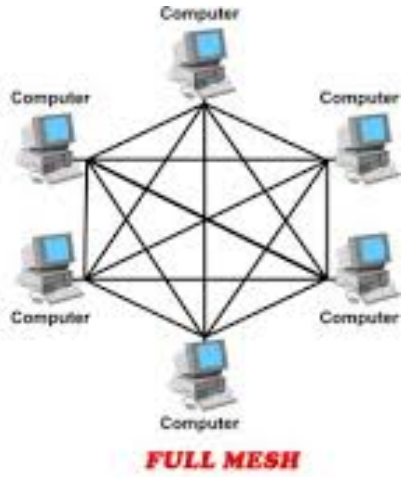


• MESH

همواره نام این توپولوژی با بحث Security **همراه است . در این توپولوژی هر نود مستقیما به همه نودها وصل میباشد .**

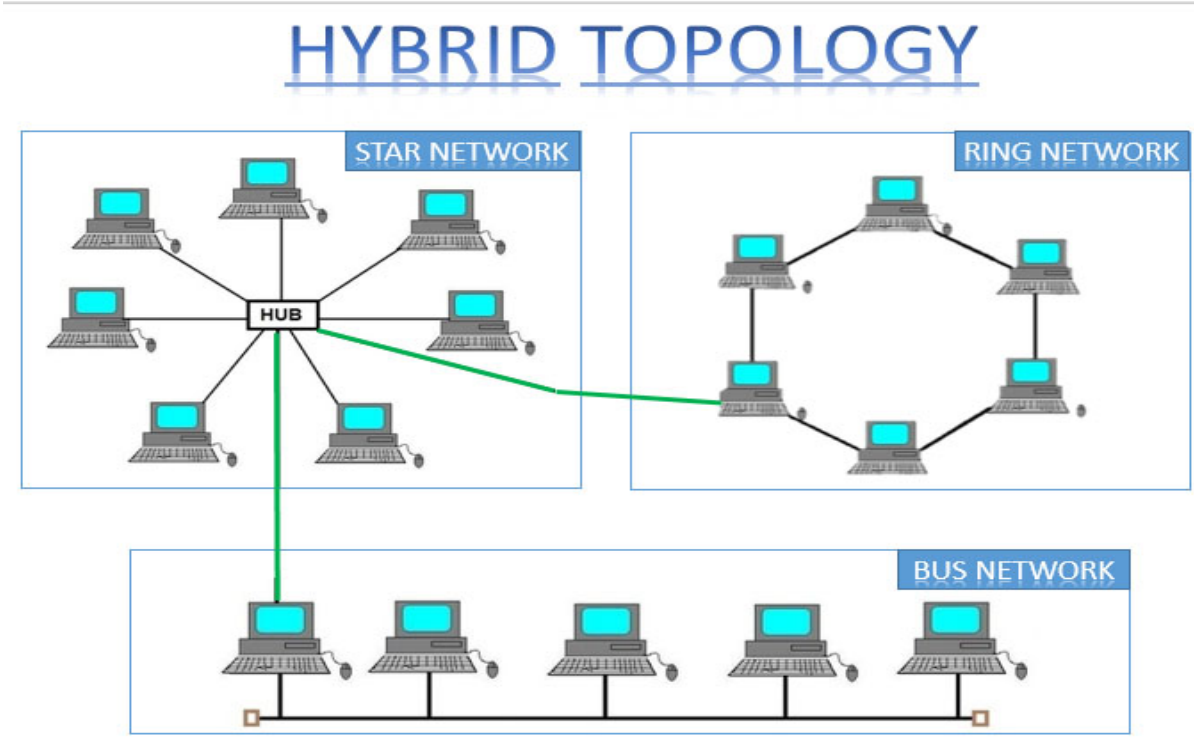
این توپولوژی درشبکه های نظامی adhoc **بسیار پر کاربرد است ، بمب های خوشه ای در جنگ از این توپولوژی استفاده می کنند هر نود (بمب) با 99 نود دیگر مستقیما در ارتباط است.**





- ***Hybrid* ترکیبی**

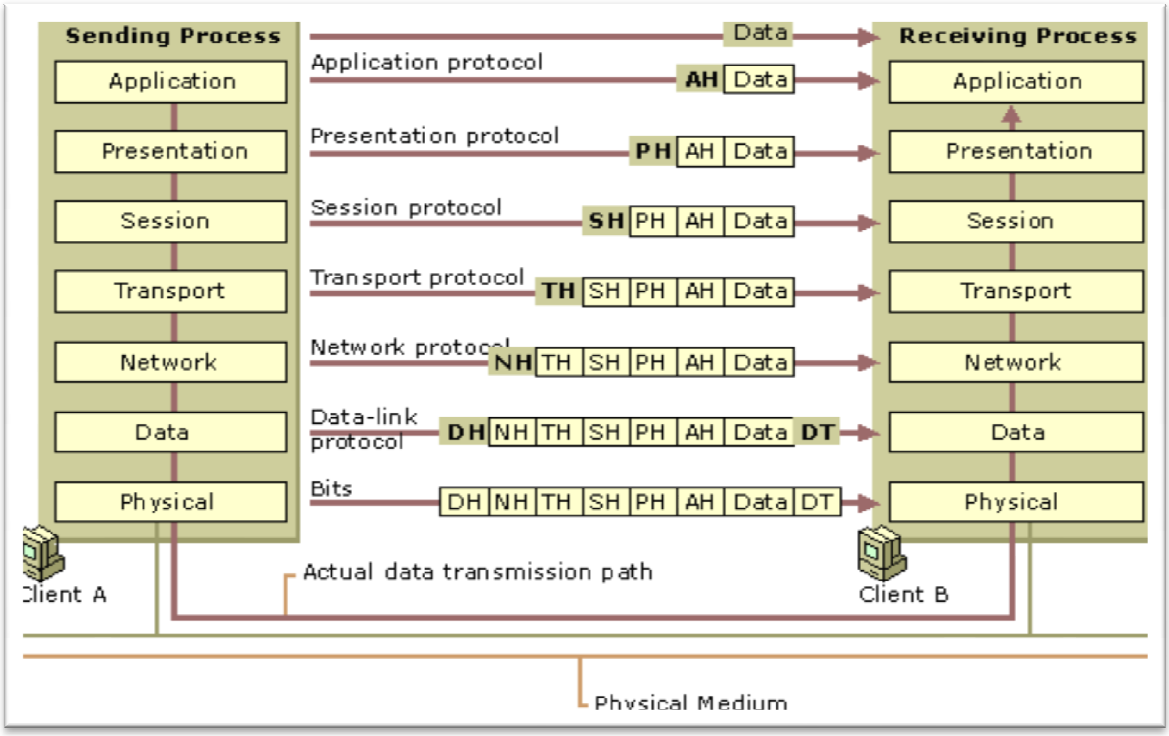
اتصال دو توپولوژی مختلف به هم ایجاد توپولوژی hybrid **میکند.**

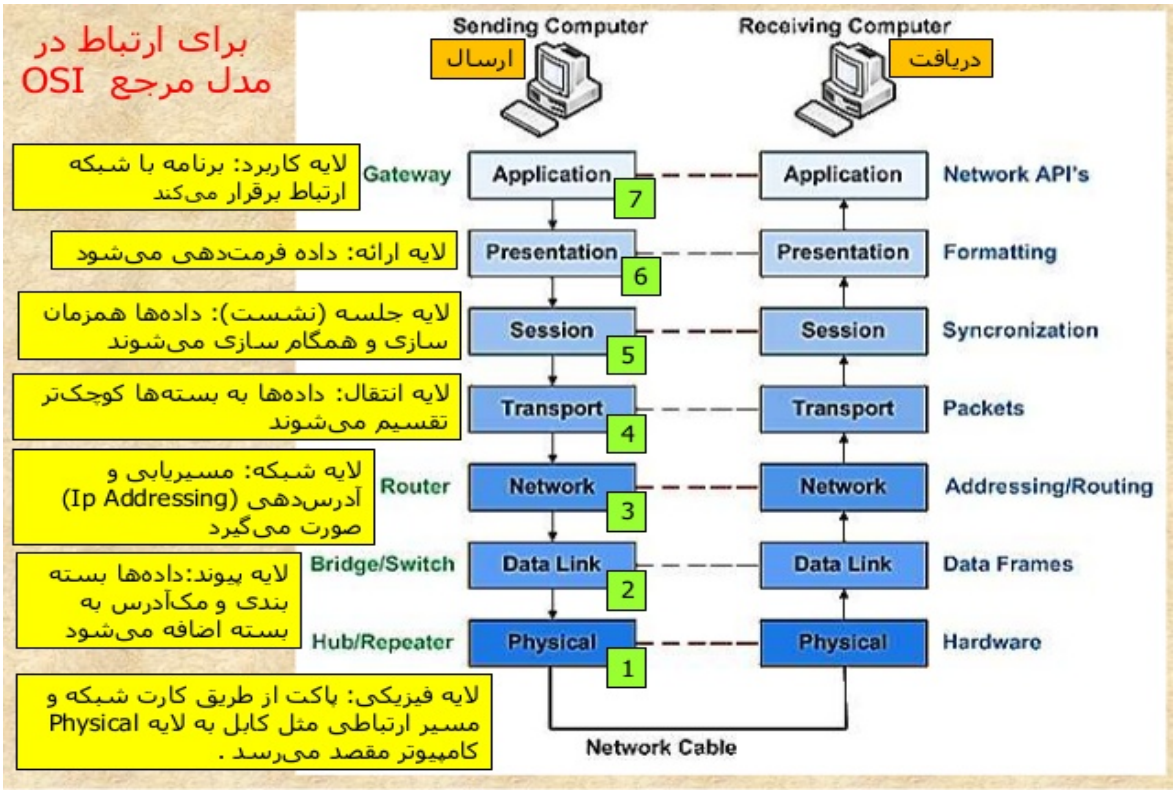


مدل OSI (Open System Interconnection) :

اساسا شبکه را لایه بندی نمودند تا وظایف هر لایه مشخص گردد تا پیاده سازی راحت تر شود ، به این صورت راحت تر میتوانیم اشکالات را پیدا کنیم .این لایه ها مبتنی بر قراردادهای هستند که سازمان جهانی ISO به عنوان مرحله ای از استانداردسازی قرادادهای لایه های مختلف ، توسعه داده اند .

OSI Model			
	Data unit	Layer	Function
Host layers	Data	7. Application	Network process to application
		6. Presentation	Data representation, encryption and decryption, convert machine dependent data to machine independent data
		5. Session	Interhost communication, managing sessions between applications
	Segments	4. Transport	Reliable delivery of segments between points on a network.
Media layers	Packet/Datagram	3. Network	Addressing, routing and (not necessarily reliable) delivery of datagrams between points on a network.
	Bit/Frame	2. Data link	A reliable direct point-to-point data connection.
	Bit	1. Physical	A (not necessarily reliable) direct point-to-point data connection.





1)لایه فیزیکی : به انتقال بیت های خام که بر روی کانال ارتباطی مربوط میشود در اینجا مدل طراحی با رابط های مکانیکی ، الکتریکی و رسانه انتقال فیزیکی سر و کار دارند .علاوه بر چگونگی اتصال 2 نود ، نوع رسانه ی انقال را مشخص میکند مثلا با کابل یا خط مخابراتی مثل leaseline**یا ماهواره ، اینکه سخت افزار چه باشد و چگونه بهم متصل بشوند(از طریق ماهواره ، فیبرنوری ، کابل ....) در این لایه تعیین میشود .**

2) لایه پیوند داده ها Data Link:

این لایه مبین نوع فرمتها است یعنی شروع و پایان فریم ، اندازه فریم ، روش انتقال فریم را مشخص میکند و وظایف این لایه شامل مدیریت فریم ، خطایابی ، ارسال مجدد فریم هاست .

یعنی شروع و پایان فریم و اندازه فریم :

مثل بسته بندی گوشتها در فریزر ، اندازه فریم مشخص میشود در واقع گوشت را باید بر اساس تعداد افراد خانواده و یا نوع استفاده (بسته های بزرگتر برای مهمانی) فریم بندی کنیم ، یعنی طراحی فریم به عهده طراح شبکه است که در شبکه فریم ها همان پکت ها هستند.

انتقال فر یم ها :

باید الگوریتم ارسال فریم مشخص شود ؛ مثلا وقتی افراد داخل کلاس بخواهد خارج شوند چند الگوریتم وجود دارد یک روش اینست که بگوییم به ترتیب سن خارج خارج شوند یا مثلا بگوییم خانم ها اول خارج شوند و ...در این جا هم برای ارسال پکت روش های مختلفی وجود دارد که طراحی schedule **ها بر عهمده طراح شبکه است مثلا ممکن است به این صورت عمل کند که اگر یک پکت به صورت** emergency **بود آن را بدون نوبت بفرست یعنی به آنها وزن و ارزش و اولویت میدهیم بنابر این دراین لایه اطلاعات به صورت فریم است طریقه و روش ارسال و انتقال فریم مبحثی است به نام زمان بندی یا** scheduling **مثلا برای ارسال 10 فریم کدام اول ارسال شود .**

در scheduling **الگوریتم های زیر وجود دارد :**

- ✓ GSS
- ✓ WRR **: الگوریتم** round robin
- ✓ STFQ
- ✓ WFQ**: این الگوریتم وزن میدهد یعنی فریم هایی کوچکترند اولویت بالاتری دارند و زودتر ارسال می شوند زیرا فریم های بزرگ ترافیک ایجاد میکند**
- ✓ WPS
- ✓ CIF-Q
- ✓ WFS
- ✓ CS-WFS
- ✓ MPFQ: multiclass priority fair queuning **درمورد ارسال داده ها به صورت عادلانه است اگر 10 تافریم برای ارسال باشند اول کوچکترها ارسال می گردد و بعد درمیان آنها اگر شبکه ترافیک نداشت فریم بزرگ تر را هم میفرستد**

*** بحث زمانبندی در تمام شبکه ها استفاده می گردد.**

3) لایه شبکه : **وظیفه این لایه مسیریابی** (Routing) **می باشد این مسیریابی عبارتند از تعیین مسیر مناسب برای انتقال اطلاعات ، لایه شبکه آدرس منطقی هر فریم را بررسی می کند و آن فریم را براساس جدول مسیریابی به مسیریاب بعدی می فرستد.**

وظیفه این لایه مسیر یابی است یعنی پیدا کردن بهترین مسیر (که می تواند کوتاهترین مسیرباشد یا پرترافیک ترین مسیر یا کم هزینه ترین مسیر است.)

برخی از Routin Protocol **ها :**

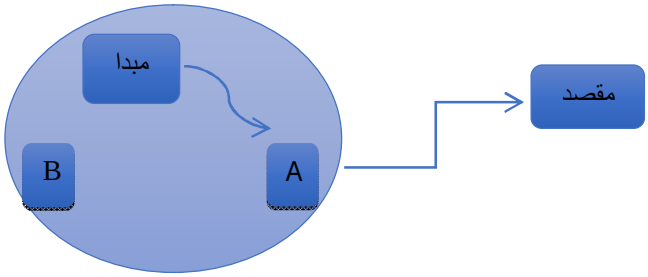
GSR Geographic Source Routing **: در شبکه** Adhoc ,MANET,VANET **داریم .**

GPSR **(**Greedy perimeter stateless routing**) : این**Routin Protocol **در همه شبکه ها وجود دارد در** Adhoc MANET,VANET, **و ...وجود دارد.(تقریب خورده است)**

GPSR**یک** Routin Protocol **مادر است و بقیه** routing protocol **ها از این گرفته شده اند و بهترین خاصیتی آن** Map Base **نبودن آن است و بر اساس** GreedyForwarding **کار میکند.**Greedy **یعنی حریصانه اما مفهموم آن به صورت زیر است :**

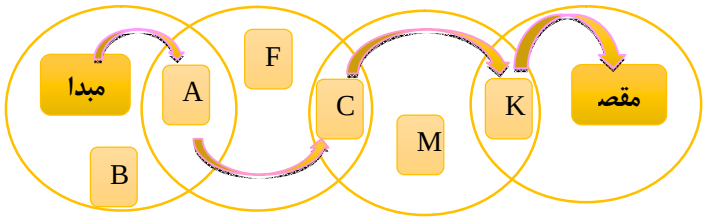
درشبکه زیر نود مبدا دارای بُرد راداری است نود A , B **در بُرد راداری مبدا هستند فرض کنید مبدا میخواهد بسته ای را به مقصد بفرستد طبق الگوریتم** Greedy forwarding **بسته به نزدیک ترین نود به مقصد که در بُرد راداری مبدا قرار دارد ارسال می گردد و آن نود بسته را به مقصد میفرستد .**

نکته : طریقه محاسبه فاصله ذکر شده در شبکه های مختلف متفاوت است مثلا در شبکه VANET **فرض بر این است که مبدا و مقصد هر دو دستگاه** GPS **دارند و هر دو** LOCATION **های خود را از** GPS **میگیرند پس مبدا** LOCATION**مقصد را دارد اما لوکیشن** Neighbor **های خود را از طریق ارسال** HELLO PACKET**در برد راداری خودش(شعاع** 1KM**) و دریافت** REPLY **آنها متوجه میشود که چه نود هایی در برد راداری اش قرار دارد.**



Greedy forwarding

تذکر : خود نود A,B **و ..هم دارای بُرد راداری هستند.**



Greedy Routing **در شبکه های حسگر بسیار کاربرد دارد و در تمام شبکه های دیگر نیز وجود دارد .**

Greedy Routing **در** MANET , ADHOC **به همین صورت عمل میکند در** VANET **هم به همین صورت عمل میکند اما در یک سناریو دچار** Problem **میشود** :

"ماشین A , B **هم جهت به سمت مقصد در حرکت اند نود** A **میخواهد بسته ی را به مقصد بفرستند اما خودش نزدیک ترین نود به مقصد است"**

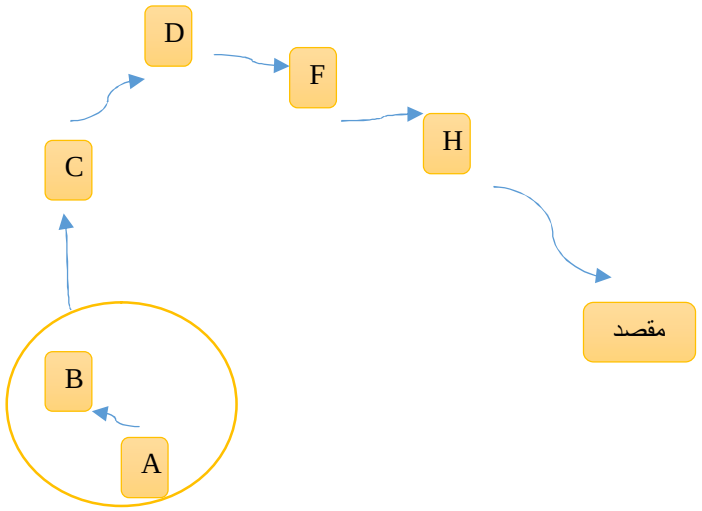
به این مشکل در شبکه های VANET **میگویند :** Local Maxima Problem

اینجاست که با مفهوم *تقریب* **آشنا میشویم :**

وقتی با مشکل بالا روبه رو میشویم میگوییم بررسی کن آیا در برد راداری آیا نود دیگری هم هست ؟ اگر بود connectivity **آن را با مقصد چک کن شاید از طریق نود های دیگر این** connectivity **به مقصد باشد.**

با این که نود B **بررسی می شود مثلا نود** B **از طریق نودهای** C,D,F,H **به مقصد** CONNECTIVITY**دارد پس بسته به** B **داده میشود تا از طریق نودهای بالا به مقصد ارسال کند.**

این راه حل تقریبی است که از شبکه های دیگر در VANET **ایجاد شده است یعنی در 90 درصد موارد مانند الگوریتم همیشگی است اما وقتی** Local Maxima Problem **ایجاد میشود** Routing **عوض میشود که به آن** *تقریب* **میگویند .**



*RoutingGPSR **الگوریتم سنگینی است چون بررسی**Connectivity **کار سختی است .**

تذکر :

در شبکه چیزی به اسم Disconnect **مطلق وجود ندارد، اساسا درصد** disconnect **یا** connect **بودن بررسی می گردد و بااحتمال اعلام می گردد مثلا فرض کنید در شکل بالا بسته تا چهار** hopcount1 **رفته ولی به** hopcount15 **نرسیده (** hopcount15 **آخرین بوده) پس شبکه روی** hopcount14 **با احتمال** 90%، disconnect **است.**

GPSR ، map-base **نیست فقط به نودهای بُرد راداری اش نگاه میکند .**

حال فرض میکنیم map **دارد بنابر این** Routing Protocol **تبدیل میشود به :**

GPCR**(**Greedy Perimeter Coordinator Routing**)**

GPCR **در واقع همان** GPSR **است اما با فرض** map-base **بودن این پروتکل تقریبی از** GPSR **است که فقط در** VANET **یا شبکه 2بعدی کاربرد دارد و در** Adhoc **و شبکه سه بعدی کاربردی ندارد . فرض کنید در بُرد راداری یک ماشین در یک 4راه دو ماشین دیگر قراردارند پس بسته را به نزدیکترین نود به 4 راه میدهد نودی که در چهارراه است را** coordinator **(هماهنگ کننده)می گویند .**

مقیاس Forwarding **در این پروتکل** Junction **(چهارراه) است این** Forwarding **براساس** Map **انجام می گردد.**

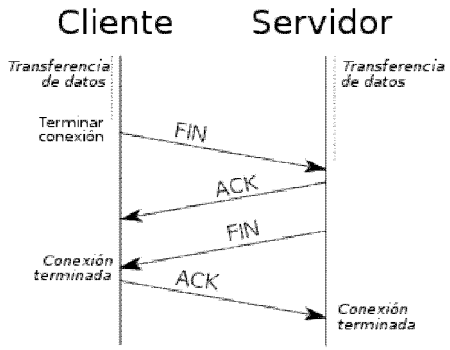
لایه 3 وظیفه روتینگ را دارد و یکی از مباحثی که مطرح است Prediction **پیشگویی است و بحث دیگری که مطرح است** movability **قابل حرکت یعنی شبکه های که حرکت دارند وبعضی ها** highmovability **هستند مانند شبکه های** AdhocVANET , FANET **این خاصیت دارد مثلا یک هواپیما جابه جایی زیادی دارد در ثانیه پس سرعت بالای دارد و وقتی آن را** Detect **میکنیم برای ارسال اطلاعات در ثانیه دیگر اونجا نیست که ارسال اطلاعات برای آن انجام بدهیم چون اطلاعات برای ثانیه قبل است همان لحظه نیست پس** Fail **می شود بنابراین مکانیزم** Prediction **مطرح میشود باید پیشگویی کنیم محاسبه انجام بدهیم مثلا براساس جهت ، مسیر باید پیشگویی کنیم الان کجا هست پس بحث**Prediction **در** Routing **شبکه های** highmovability **(سرعت بالا مثل** FANET **) خیلی مطرح است .**

4) لایه انتقال Transport :

وظیفه ارسال مطمئن یک فریم به مقصد را به عهده دارد لایه انتقال پس از ارسال یک فریم به مقصد منتظر می ماند تا سیگنالی از مقصد مبنی بر دریافت آن فریم دریافت کند درصورتی که در این لایه مقصدپیام خود را دریافت نکند مبدا مجددا اقدام به ارسال همان فریم به مقصد میکند ، بحثی که در این لایه مطرح است طریقه انتقال به صورت UDP **یا** TCP **است .**

به طور کلی در همه شبکه از جمله VANET , MANET , ADHOC**و ...تنها دو نوع** Connection **داریم :**

1) *Connection Oriented* ***یا اتصال گرا*** *(TCP)* **:در این کانکشن اول اتصال برقرار میشود به عنوان مثال فرض کنید در دو طرف سیستم تلفن ثابت وقتی هردو طرف برای اولین بار صدای هم را میشنوند یعنی کانکشن برقرار شده است در مرحله بعد یکی یک طرف سه پکت : "سلام" "خوبی؟" "چه خبر؟" را ارسال میکند و مقصد هم این سه پکت را با همین ترتیب دریافت میکند پس اولین خاصیت** Connection Oriented **در ارسال بسته های اطلاعاتی مطابق بودن ترتیب ارسال از مبدا و ترتیب دریافت در مقصد است . دومین خاصیت** Connection Oriented **این است که هر بسته ای که به مقصد میرسد پیام** acknowledge **را به مبدا ارسال میکند.**



2) *Connection less* ***یا غیر اتصال گرا*** *(UDP) :*

در این نو کانکشن ، Acknowledge **وجود ندارد همچنین هیچ تضمینی وجود ندارد که بسته ای که اول ارسال شده و روی رسانه انتقال قرار گرفته است در مقصد نیز اول دریافت گردد برای این موضوع میتوان مثال اداره پست را در نظر گرفت که نامه ها در زمانهای مختلف می رسند که به دلیل این است که روتینگ های مختلفی ممکن است وجود داشته باشد . برای** Connection **های شبکه** highmovability **استفاده میشود .**

* * UDP **مخصوص شبکه های** highmovability **(شبکه های پر سرعت مانند** FANETD**یا** VANET**)**

* * highmovability **چیست :** characteristic **است مثل شبکه های** Adhoc ,VANET **وقتی یک شبکه را تعریف میکنیم باید ابتدا** characteristic **آنرا تعریف کنیم مثلا** Time mobility

characteristic **هر شبکه با شبکه بعدی فرق میکند . وقتی این موارد مطالعه میکنیم ذهن باز میشود مثلا درمورد** SensorNetwork **میخوانیم یک مشخصه داریم به نام** Power saving or energy saving **که در** VANET **این پارامترنداریم .**

* * **هسته لایه 4 ،** Connection **های** TCP**/**UDP **است .**

5) لایه اجلاس session: **وظیفه برقراری یک ارتباط منطقی بین نرم افزارهای دو کامپیوتری که به یکدیگر متصل هستند به عهده این لایه است وقتی که یک ایستگاه بخواهد به یک سرویس دهنده متصل شود، سرویس دهنده فرآیند برقراری ارتباط را بررسی می کند سپس از ایستگاه درخواست نام کاربر رمز عبور راخواهد کرد این فرآیند نمونه ای از یک اجلاس است .**

6) لایه نمایش : این لایه اطلاعات را از لایه کاربرد دریافت نموده و آنها را به شکل قابل فهم برای کامپیوتر مقصد تغییر می دهد .

7) لایه کاربرد : این لایه امکان دسترسی کاربران به شبکه را با استفاده از نرم افزارهایی مثل FTP **و ایمیل فراهم می سازد .**

فصل دوم

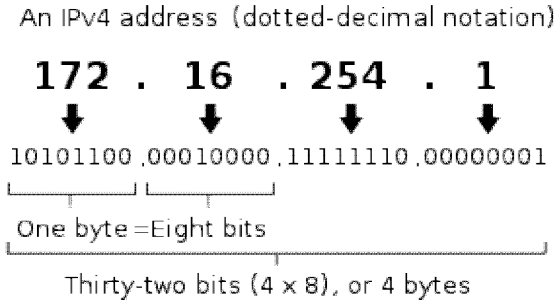
IP ADDRESSING
&
SUBNET TING

IP addresses and Subnet ting objectives

- IP addresses Introduction
- Network & Host
- IP addresses Classes
- Network ID & broadcast address
- subnet ting
- Supernetting

مقدمه :

IP مورد بحث ما در این درس IP ورژن چهارم میباشد (IPV4) که از چهار اوکتت هشت بیتی تشکیل شده است:



*** طریقه بدست آوردن باینری یک عدد دهدهی :**

با استفاده از جدول از پیش طراحی شده که در ردیف بالای آن وزن های عددی از 1 تا 128 چیده شده است بایستی نزدیک ترین عدد به عدد مورد نظر که میخواهیم آن را به باینری 32 بیتی تبدیل کنیم ، پیدا کنیم ، به عنوان مثال میخواهیم باینری عدد 38 را پیدا کنیم در بین وزن ها نزدیک ترین عدد 32 می باشد در مرحله بعد 32 باید با چه عددی جمع شود تا حاصل 38 شود ؟ ، واضح است که با جمع دو عدد 4 و 2 . به این ترتیب زیر وزن های 32 ، 4 ، 2 عدد باینری 1 و زیر باقی اعداد صفر قرار میدهیم در نهایت میبینیم که این تبدیل انجام شده است.

	128	64	32	16	8	4	2	1
38	0	0	1	0	0	1	1	0
68	0	1	0	0	0	1	0	0
192	1	1	0	0	0	0	0	0

تعاریف :

IP Address

IP Address به عنوان شناسه در شبکه ها استفاده می شود که می تواند نشانگر یک نود شبکه و یا آدرس شبکه باشد. IP Address ، آدرس Logical لایه Network در مدل OSI است.
IP Address ها ، بصورت دسیمال، به صورت ۴ عدد (هرکدام از ۰ تا ۲۵۵ می تواند مقدار گیرد) ، که هر کدام با نقطه از هم جدا شده اند ؛ نوشته می شود.
به عنوان مثال :

192.168.0.1

نمایانگر یک آدرس IP است.
ساختار IP Address ها بصورت یک عدد باینری ۳۲ بیتی (۴ بایتی) است که هر ۸ بیت (۱ بایت) توسط نقطه از دیگر بیتها (بایتها) جدا می شود.
به عنوان مثال :

11000000. 10101000. 00000000. 00000001

نمایانگر یک آدرس IP است.

Network ID و Broadcast Address

- هر Range آدرس IP داخل یک شبکه واحد، مجموعه ای از یکسری آدرس IP است که همگی داخل آن شبکه اند . از این مجموعه، دو آدرس منحصر به فرد وجود دارد:
- آدرس Network ID: که مشخصه و معرف آن شبکه است.
- آدرس Broadcast: که برای دسترسی به همه نودهای آن شبکه استفاده می شود.
- نکته: این دو آدرس را نمی توان به عنوان آدرس معتبر، به نودها اختصاص داد.

معرفی Host و Network

Borhani Hasan
Hosseini Neda
Sajadi Pedram
Borhani Ali
Borhani Mohammad
Hosseini Siavash
Sajadi Shabnam

معرفی Host و Network

Borhani	Hasan
Hosseini	Neda
Sajadi	Pedram
Borhani	Ali
Borhani	Mohammad
Hosseini	Siavash
Sajadi	Shabnam

معرفی Host و Network

Network	Host
Borhani	Hasan
Hosseini	Neda
Sajadi	Pedram
Borhani	Ali
Borhani	Mohammad
Hosseini	Siavash
Sajadi	Shabnam

معرفی Host و Network

Network	Host
Borhani	Hasan
Borhani	Ali
Borhani	Mohammad

Network	Host
Hosseini	Neda
Hosseini	Siavash

Network	Host
Sajadi	Pedram
Sajadi	Shabnam

معرفی Host و Network

13	.21.48.25
Network	Host

192.168.68.	32
Network	Host

172.16	115.2
Network	Host

در مثال های بالا چگونه متوجه شدیم که مرز بین Host, Network **کجاست؟ به عنوان مثال چرا برای** ip **اول یعنی :** 13.21.48.25 **مرز به این صورت مشخص شده است که اوکتت اول را** Network **و سه اوکتت بعد را** Host **در نظر گرفتیم ؟**

برای فهمیدن مرز بین این دو باید کلاس های IP **را بشناسیم .**

کلاس A : **کلاسی است که اوکت اول** IP **بین 1 تا 126 میباشد و قانون این کلاس به این صورت است که اوکتت اول میشود** Network **و سه اوکتت بعدی** Host **در نظر گرفته میشود.**

کلاس B : **کلاسی است که اوکت اول** IP **بین 128 تا 191 میباشد و قانون این کلاس به این صورت است که دو اوکتت اول میشود** Network **و دو اوکتت بعدی** Host **در نظر گرفته میشود.**

کلاس C : **کلاسی است که اوکت اول** IP **بین 192 تا 223 میباشد و قانون این کلاس به این صورت است که سه اوکتت اول میشود** Network **و اوکتت آخر** Host **در نظر گرفته میشود.**

انواع کلاس های IP Address

Class A

Network	Host		
1 to 126	x	x	x

Class B

Network		Host	
128 to 191	x	x	x

Class C

Network			Host
192 to 223	x	x	x

آیا آدرس های 220.34.30.42و 220.34.32.48 در یک شبکه قرار دارند ؟

Class ? **C**

Network	Host
220.34.30.	42

Network	Host
220.34.32.	48

آیا قسمت Network یکسان است ؟ **خیر**

پس دو آدرس فوق داخل یک شبکه قرار ندارند.

*** البته این بررسی ناقص است و در ادامه بحث خواهیم گفت که علاوه بر بررسی موارد بالا باید Subneting نیز بررسی شود ***

*طریقه به دست آوردن NetworkID&Broadcast address :

فرض کنید شخصی وارد یکی از اتاق های دانشگاه میشود و میخواهد از طریق کامپیوتری که آنجا وجود دارد ویروسی را به تمام سیستم های دانشگاه بفرستد ، ایتدا ip **کامپیوتری که در اختیار دارد را به دست می آورد که به عنوان مثال** 80.32.51.60 **میباشد حال باید دو کار انجام دهد :**

1) IP **کل دانشگاه را بدست بیاورد (**IP **مادر یا** NetID **)**

2) آی پی Broadcast **که به وسیله آن ویروس را به تمام سیستم ها بفرستد.**

قانون به دست آوردن NetworkID **:**

- **ابتدا باینری** IP **مورد نظر را از طریق روش گفته شده بدست می آوریم.**
- **قسمت** NET **بدون تغییر نوشته میشود و قسمت** HOST **همگی صفر میشود.**
- **حال این 32 بیت را به دهدهی تبدیل میکنیم.**

قانون به دست آوردن Broadcast Address **:**

- **ابتدا باینری** IP **مورد نظر را از طریق روش گفته شده بدست می آوریم.**
- **قسمت** NET **بدون تغییر نوشته میشود و قسمت** HOST **همگی یک میشود.**
- **حال این 32 بیت را به دهدهی تبدیل میکنیم.**

NetID و Broadcast address شبکه ای که آدرس IP ، 80.32.51.60 در آن وجود دارد را پیدا کنید.

Class ? A

Network	Host
80.	32.51.60

Network	Host
01010000 .	00100000 . 00110011 . 00111100

Network ID :

Network	Host
01010000 .	00000000 . 00000000 . 00000000

پس NetID این شبکه ، می شود : 80.0.0.0

Network	Host
80.	32.51.60

Network	Host
01010000 .	00100000 . 00110011 . 00111100

Broadcast Address :

Network	Host
01010000 .	11111111 . 11111111 . 11111111

پس Broadcast Address این شبکه ، می شود : 80.255.255.255

مثال :

شبکه ای که آدرس IP ، 201.202.32.40 در آن وجود دارد را **تحلیل کنید**.

Objectives	Value
Class	?
Network ID	?
First IP address	?
Last IP address	?
Broadcast address	?
Number of Available IP addresses	?

Class ? **C**

Network	Host
201.202.32	.40

Network	Host
201.202.32	.40

Network	Host
11001001 . 11001010 . 00100000	. 00101000

Network ID :

Network	Host
11001001 . 11001010 . 00100000	. 00000000

پس **Net ID** این شبکه ، می شود : **201.202.32.0**

Objectives	Value
Class	C
Network ID	201.202.32.0
First IP address	?
Last IP address	?
Broadcast address	?
Number of Available IP addresses	?

Network	Host
201.202.32	.40

Network	Host
11001001 . 11001010 . 00100000	. 00101000

Broadcast Address :

Network	Host
11001001 . 11001010 . 00100000	. 11111111

پس Broadcast Address این شبکه ، می شود : 201.202.32.255

Objectives	Value
Class	C
Network ID	201.202.32.0
First IP address	?
Last IP address	?
Broadcast address	201.202.32.255
Number of Available IP addresses	?

Network	Host
201.202.32	.40

Network	Host
11001001 . 11001010 . 00100000	. 00101000

اولین آدرس قابل استفاده شبکه :

Network	Host
11001001 . 11001010 . 00100000	. 00000001

Objectives	Value
Class	C
Network ID	201.202.32.0
First IP address	201.202.32.1
Last IP address	?
Broadcast address	201.202.32.255
Number of Available IP addresses	?

Network	Host
201.202.32	.40

Network	Host
11001001 . 11001010 . 00100000	. 00101000

آخرین آدرس قابل استفاده شبکه :

Network	Host
11001001 . 11001010 . 00100000	. 11111110

پس اولین آدرس IP این شبکه ، می شود : 201.202.32.1

Network	Host
201.202.32	.40

Network	Host
11001001 . 11001010 . 00100000	. 00101000

آخرین آدرس قابل استفاده شبکه :

Network	Host
11001001 . 11001010 . 00100000	. 11111110

پس آخرین آدرس IP این شبکه ، می شود : 201.202.32.254

Objectives	Value
Class	C
Network ID	201.202.32.0
First IP address	201.202.32.1
Last IP address	201.202.32.254
Broadcast address	201.202.32.255
Number of Available IP addresses	?

Network	Host
201.202.32	.40

Network	Host
11001001 . 11001010 . 00100000	. 00101000

تعداد آدرس های IP قابل استفاده در این شبکه :

Network	Host
11001001 . 11001010 . 00100000	. 00101000

فرمول تعداد IP های قابل تعریف درIP شبکه ی 201.202.32.40 :

$2^n - 2$

تعداد بیت های هاست

اولین IP وآخرین IP
- IPاول : تمام HOST صفر شود (NetID)
- IP دوم: Broadcast address

پس تعداد آدرس های IP قابل استفاده در این شبکه ، $2^8 - 2 = 254$ است.

جواب نهایی تحلیل :

Objectives	Value
Class	C
Network ID	201.202.32.0
First IP address	201.202.32.1
Last IP address	201.202.32.254
Broadcast address	201.202.32.255
Number of Available IP addresses	254

Subnet Mask

مقدمه :

SubnetMask **یعنی شکستن یک شبکه بزرگ به شبکه های کوچکتر و عمل** subnetting **یعنی شکستن** IP**. در شبکه** 201.202.32.40 **تعداد** 254 pc **می توانند شبکه شوند تحت** subnetmask:255.255.255.0

اما اگر شبکه فوق به چند زیر شبکه بشکند ، subnetmask **آن تغییر میکند .کامپیوتر برای تشخیص تعلق یا عدم تعلق دو آدرس** IP **به یک شبکه از مفهومی به نام** Subnet Mask **استفاده میکند. به این صورت که تمام بیتهای** Network **را یک و تمام بیتهای** Host **را صفر در نظر میگیرد تا** Subnet mask **را بسازد. سپس** Subnet Mask **را در آدرس** IP**،** AND **میکند.**

Subnet Mask Class A

Network	Host
11111111 .	00000000 . 00000000 . 00000000

255 . 0 . 0 . 0

Subnet Mask Class B

Network	Host
11111111 . 11111111 .	00000000 . 00000000

255 . 255 . 0 . 0

Subnet Mask Class C

Network	Host
11111111 . 11111111 . 11111111 .	00000000

255 . 255 . 255 . 0

اگر کامپیوتر بخواهد تشخیص دهد که دوآدرس در یک شبکه قرار دارد یا خیر به این ترتیب عمل می کنیم

Subnet Mask1 AND IP 1 = Result 1

Subnet Mask2 AND IP 2 = Result 2

IF Result1 = Result2 then

IP1 & IP2 are in the same Network.

IF Not then

IP1 & IP2 are in the different Networks.

مثال :

آیا آدرس های 80.23.45.2 و 80.24.35.1 در یک شبکه اند ؟

80.23.45.2
Class ? **A**
Subnet Mask Class A ? **255.0.0.0**
80.23.45.2 AND 255.0.0.0 = Result1

01010000.00010111.00101101.00000010
AND
11111111.00000000.00000000.00000000

01010000.00000000.00000000.00000000

80.24.35.1
Class ? **A**
Subnet Mask Class A ? **255.0.0.0**
80.24.35.1 AND 255.0.0.0 = Result2

01010000.00011000.00100011.00000001
AND
11111111.00000000.00000000.00000000

01010000.00000000.00000000.00000000

Result1 = 01010000.00000000.00000000.00000000
Result2 = 01010000.00000000.00000000.00000000

بنابراین آدرس های 80.23.45.2 و 80.24.35.1 در یک شبکه اند.

Special IP Addresses

127.0.0.0	(loop back)
224.0.0.0 to 238.0.0.0	(Multicast)
239.0.0.0 to 254.0.0.0	(Experimental)
0.0.0.0 & 255.0.0.0	(Reserved)

Private IP Addresses

Class A
10.0.0.0
255.0.0.0
Class B
172.16.0.0 to 172.31.0.0
255.255.0.0
Class C
192.168.0.0
255.255.255.0

حالا این سوال پیش می آید وقتی می توان از روی خود آدرس تشخیص داد که کدام قسمت مربوط به Network **و کدام قسمت مربوط به** Host **است. پس لزوم استفاده از** Subnet Mask **چیست؟**

Subnetting

تا اینجا در مورد شبکه هایی صحبت شد که آدرس دهی آنها با کلاس بود اما ما روش دیگری برای آدرس دهی داریم که به آن آدرس دهی بی کلاس (Class Less) گفته می شود در چنین مواردی تشخیص این مطلب که کدام قسمت مربوط به Network و کدام قسمت مربوط به Host است، کار مشکلی است، برای تشخیص network و host استفاده از مفهوم Subnet Mask ضروری است.

می توان گفت که عدد Subnet Mask هیچ ارتباطی با آدرس IP ندارد و فقط مشخص کننده این است که کدام قسمت آدرس IP مربوط به Network و کدام قسمت مربوط به Host است.

- هدف از Subnet ting این است که یک Range از آدرس های IPرا که به ما تعلق دارد، به چند Rangeآدرس مجزا تقسیم کنیم تا بتوانیم از هر Range جداگانه استفاده کنیم. مثلا ممکن است بخواهیم برای کاهش ترافیک، شبکه را به چند سگمنت، تقسیم کنیم و بین سگمنت ها روتر قرار دهیم.
- وقتی یک شبکه را Subnet می کنیم، Subnet Mask جدیدی خواهیم داشت که معرف بخش Network و Host خواهد بود.

Network	Host
Shariaty	Seyed Ali
Shariaty	Haj Hossein
Shariaty	Seyed Arman
Shariaty	Haj Ebrahim
Shariaty	Seyed Naser

زمانی که Subnet Mask بصورت بالا در نظر گرفته می شود، همه عضو یک خانواده حساب می شوند.

Network →	Host
Shariaty	Seyed Ali
Shariaty	Haj Hossein
Shariaty	Seyed Arman
Shariaty	Haj Ebrahim
Shariaty	Seyed Naser

حالا فرض کنید خط Subnet Mask
(خط مجزا کننده Network از Host) را به سمت راست بکشیم:

Network	Host
Shariaty Seyed	Ali
Shariaty Haj	Hossein
Shariaty Seyed	Arman
Shariaty Haj	Ebrahim
Shariaty Seyed	Naser

Network	Host
Shariaty Seyed	Ali
Shariaty Haj	Hossein
Shariaty Seyed	Arman
Shariaty Haj	Ebrahim
Shariaty Seyed	Naser

می بینید که دیگر همه عضو یک خانواده نیستند یک خانواده بزرگ را بـه دو خـانواده کوچک تقسیم کردیم.

Network	Host
Shariaty Seyed	Ali
Shariaty Haj	Hossein
Shariaty Seyed	Arman
Shariaty Haj	Ebrahim
Shariaty Seyed	Naser

Network	Host
Shariaty Seyed	Ali
Shariaty Haj	Hossein
Shariaty Seyed	Arman
Shariaty Haj	Ebrahim
Shariaty Seyed	Naser

یک خانواده بزرگ را به دو خانواده کوچک تقسیم کردیم.
چگونه ؟ با افزایش فضای Network و در نتیجه کاهش فضای Host

حالا با دیدی که از این مثال به دست آوردیم، متوجه می شویم که برای اینکه یک Network بزرگ را به چند Network کوچکتر تقسیم کنیم، راه حل این است که قسمت Network را بزرگتر و در نتیجه Host را کوچکتر کنیم. اینکار با قرض کردن بیت های قسمت Host و اضافه کردن این بیت ها به قسمت Network صورت می گیرد.

حالا سوال این است که چند بیت باید از host قرض بگیریم و به network اضافه کنیم؟

اگر تعداد Subnet های مورد نیاز ، مطرح بود:

$$2^n \geq \text{Number of Subnets Needed}$$

n تعداد بیت هائی است که باید از سمت راست از Host قرض بگیریم و به Network اضافه کنیم.

اگر تعداد حداقل IP های قابل استفاده ، مورد نظر ما بود :

$$2^h - 2 \geqslant \text{Number of available IP addressees}$$

h تعداد بیتهائی است که از سمت چپ Host می شماریم تا تعداد بیت هائی که باید قرض بگیریم ، مشخص شود.

شبکه 10.0.0.0 متعلق به سازمان ما می باشد. می خواهیم آنرا طوری تقسیم کنیم که ۵ شبکه مجزا به ما بدهد.

Class ? A

Network	Host
00001010 .	00000000 . 00000000 . 00000000

$$2^n \geqslant 5 \qquad n = 3$$

Network	→	Host
00001010 .		00000000 . 00000000 . 00000000

Network	Host
00001010 . 000	00000 . 00000000 . 00000000

گفتیم n تعداد بیت هایی است که باید از سمت host قرض بگیریم و به network اضافه کنیم بنابر این در این مثال network برابر با 11 بیت میشود در نتیجه فضای host کاهش و فضای network افزایش پیدا کرد.در مرحله بعد باید subnet mask جدید را پیدا کنیم برای این منظور قسمت network همگی 1 و host همگی صفر میشوند . نکته ای که وجود دارد این است که که طریقه خواندن 8 بیتی است بنابر این بیت ها در subnet mask جدید هم به صورت هشت بیتی خوانده میشود :

New Subnet Mask ?

Network	Host
11111111 . 111	00000 . 00000000 . 00000000

255.224.0.0 **OR** **/11**

/11 : دو معنی میدهد 1) یعنی قسمت network شامل 11 بیت است و 2) به معنی 255.224.0.0 (اگر 11 بیت را یک کنیم و host را کامل صفر کنیم میشود 255.224.0.0)

اکنون new subnet mask برابر با 255.224.0.0 است پس میتوانیم شبکه ی 10.0.0.0 را به 2^3 = 8 شبکه مستقل بشکنیم و چون در صورت مساله از ما خواسته 5 شبکه بنابر این 5 شبکه از 8 شبکه را استفاده میکنیم.

حالا تمام حالت هایی را که با تغییر Network جدید (سه بیت اضافه شده) به دست می آوریم، حساب می کنیم

Network	Host
00001010 . 000	00000 . 00000000 . 00000000
00001010 . 001	00000 . 00000000 . 00000000
00001010 . 010	00000 . 00000000 . 00000000
00001010 . 011	00000 . 00000000 . 00000000
00001010 . 100	00000 . 00000000 . 00000000
00001010 . 101	00000 . 00000000 . 00000000
00001010 . 110	00000 . 00000000 . 00000000
00001010 . 111	00000 . 00000000 . 00000000

10.0.0.0 /11
10.32.0.0 /11
10.64.0.0 /11
10.96.0.0 /11
10.128.0.0 /11
10.160.0.0 /11
10.192.0.0 /11
10.224.0.0 /11

اگر بخواهیم شبکه سوم را آنالیز می کنیم به این ترتیب عمل می کنیم

شبکه 10.64.0.0/11 را تحلیل کنید.

Network	Host
00001010 . 010	00000 . 00000000 . 00000000

Network ID :

Network	Host
00001010 . 010	00000 . 00000000 . 00000000

پس NetID می شود : **10.64.0.0**

Broadcast address:

Network	Host
00001010 . 010	11111 . 11111111 . 11111111

پس Broadcast address می شود : **10.95.255.255**

Network	Host
00001010 . 010	00000 . 00000000 . 00000000

First available IP

Network	Host
00001010 . 010	00000 . 00000000 . 00000001

پس اولین آدرس IP قابل استفاده در این شبکه می شود :

10.64.0.1

Last available IP address:

Network	Host
00001010 . 010	11111 . 11111111 . 11111110

پس آخرین آدرس IP قابل استفاده در این شبکه می شود :

10.95.255.254

Network	Host
00001010 . 010	00000 . 00000000 . 00000000

تعداد بیت های هاست

$2^{21}-2$

تعداد آدرس IP قابل استفاده در این شبکه :

می شود ۲۰۹۷۱۵۰ آدرس IP

Subnet Mask	255.224.0.0
Network ID	10.64.0.0
First IP address	10.64.0.1
Last IP address	10.95.255.254
Broadcast address	10.95.255.255
Number of Available IP addresses	2097150

توضیح تکمیلی :

یک network را به 8 شبکه کوچکتر شکستیم که با آنالیز تنها یکی از آنها به این نتیجه رسیدیم که تنها یکی از این 8 شبکه میتوانیم 2097150 آی پی مختلف (کامپیوتر) تعریف کنیم !

شبکه 172.16.0.0 را طوري Subnet کنید که در هر شبکه جدید ۳۰۰ آدرس IP قابل استفاده وجود داشته باشد.

Class ? **B**

Network	Host
10101100 . 00010000 .	00000000 . 00000000

$$2^h - 2 \geqslant 300$$

$$h = 9 \qquad n = 7$$

Network →	Host
10101100 . 00010000 .	00000000 . 00000000

Network	Host
10101100 . 00010000 . 0000000	0 . 00000000

Network	Host
10101100 . 00010000 . 0000000	0 . 00000000

New Subnet Mask ?

Network	Host
11111111 . 11111111 . 1111111	0 . 00000000

255.255.254.0 OR **/23**

توضیح تکمیلی : 7 بیت به host **منتقل کرده بودیم بنابر این داریم** $2^7 = 128$ **، یعنی به 128** network **شکسته شده است که در هر** network **تعداد 300 عدد** IP **وجود دارد .**

حالا حالت هایی را که با تغییر Network جدید (۷ بیت اضافه شده) به دست می آوریم را حساب می کنیم

Network	Host
10101100 . 00010000 . 0000000	0 . 00000000
10101100 . 00010000 . 0000001	0 . 00000000
10101100 . 00010000 . 0000010	0 . 00000000
10101100 . 00010000 . 0000011	0 . 00000000
…	…

Network	Host
10101100 . 00010000 . 0000000	0 . 00000000
10101100 . 00010000 . 0000001	0 . 00000000
10101100 . 00010000 . 0000010	0 . 00000000
10101100 . 00010000 . 0000011	0 . 00000000
…	…

172.16.0.0/23
172.16.2.0/23
172.16.4.0/23
172.16.6.0/23
…

اگر بخواهیم شبکه دوم را آنالیز می کنیم به این ترتیب عمل می کنیم

شبکه 172.16.2.0/23 را تحلیل کنید.

Network	Host
10101100 . 00010000 . 0000001	0 . 00000000

Network ID :

Network	Host
10101100 . 00010000 . 0000001	0 . 00000000

پس NetID می شود : **172.16.2.0**

Broadcast address:

Network	Host
10101100 . 00010000 . 0000001	1 . 11111111

پس Broadcast address می شود : **172.16.3.255**

Network	Host
10101100 . 00010000 . 0000001	0 . 00000000

First available IP

Network	Host
10101100 . 00010000 . 0000001	0 . 00000001

پس اولین آدرس IP قابل استفاده در این شبکه می شود :

172.16.2.1

Last available IP address:

Network	Host
10101100 . 00010000 . 0000001	1 . 11111110

پس آخرین آدرس IP قابل استفاده در این شبکه می شود :

172.16.3.254

Network	Host
10101100 . 00010000 . 0000001	0 . 00000000

تعداد آدرس IP قابل استفاده در این شبکه : $2^9 - 2 = 510$

Supernetting

دقیقا عمل عکس Subnet ting است. یعنی چند شبکه کوچک را با هم ادغام کرده و یک شبکه بزرگ ایجاد می کنیم. از این مورد برای کاهش ترافیک شبکه هایی که قرار است روتر بسته ای را به چند شبکه بفرستد استفاده می شود.

نکته مهم این است که هر دو شبکه ای را نمی توان با هم Supernet کرد.

چهار شبکه با IP های زیر وجود دارد ، میخواهیم این چهار شبکه را ادغام کنیم به یک شبکه مادر :

10.128.0.0 /11
10.160.0.0 /11
10.192.0.0 /11
10.224.0.0 /11

ابتدا آدرس ها را بصورت باینری می نویسیم:

Network	Host
00001010 . 100	00000 . 00000000 . 00000000
00001010 . 101	00000 . 00000000 . 00000000
00001010 . 110	00000 . 00000000 . 00000000
00001010 . 111	00000 . 00000000 . 00000000

10.128.0.0 /11
10.160.0.0 /11
10.192.0.0 /11
10.224.0.0 /11

Network ←	Host
00001010 . 100	00000 . 00000000 . 00000000
00001010 . 101	00000 . 00000000 . 00000000
00001010 . 110	00000 . 00000000 . 00000000
00001010 . 111	00000 . 00000000 . 00000000

همانطور که مشاهده میکنید تا بیت شماره 9 هر چهار شبکه مشابه هم هستند ولی دو بیت آخر مشابه نیستند بنا بر این دو بیت آخر را به Host **شیفت میدهیم ، قبلا** /11 **بود و اکنون به** /9 **تبدیل میشود :**

Network	Host
00001010 . 1	0000000 . 00000000 . 00000000
00001010 . 1	0100000 . 00000000 . 00000000
00001010 . 1	1000000 . 00000000 . 00000000
00001010 . 1	1100000 . 00000000 . 00000000

New Subnet Mask :

Network	Host
11111111 . 1	0000000 . 00000000 . 00000000

/9 OR 255.128.0.0

Network ID:

Network	Host
00001010 . 1	0000000 . 00000000 . 00000000

10.128.0.0

فصل سوم

مهندسی ترافیک

مفهوم *request* : به معنای درخواست است .

مفهوم *reply* : به معنای پاسخ است .

مثال : وارد کردن آدرس URL به معنی request و باز شدن کامل WEBPAGE به معنی reply میباشد .

مفهوم *Transaction* :یعنی reply+request هر transaction ای دارای محیط عملیاتی است مثلا استاد از دانشجویی معدلش را می پرسد و دانشجو پاسخ می دهد (یک Transaction اتفاق افتاده) که محیط عملیاتی آموزشی است .

هر Transaction یک محیط عملیاتی دارد مانند Transaction در دیتابیس ، Query است .

بستگی به Transaction یک سیستم معماری های متفاوتی خواهیم داشت جنس Transaction ها باهم متفاوت هست .

مفهوم Performance : دارای پارامتر است به معنی بهره وری است یعنی با کم و زیاد کردن پارامترها میتوان بهره وری را تغییر دهیم پارامترهای آن عبارتند از :

1) **Throughput : ترجمه آن گذردهی است به معنی ورودی سیستم است مثل تعداد بازدیدکننده های سایت معنی throughput در وب سایت : تعداد کلیک های که آدرس URL وب سایت را می بیند .**
یا به عنوان مثالی دیگر فرض کنید شخصی را مامور میکنیم که جلوی درب ورودی بانک بایستد و وظیفه اش شمارش ورود و خروج تعداد اشخاصی باشد که از 8 صبح تا 4 بعد از ظهر به بانک مراجعه کردند در نهایت این مامور به ما این عدد را مثلا 450 اعلام میکند بنابراین Throughput این بانک در این بازه زمانی برابر با 450 میباشد . هر چه Throughput بالاتر رود ترافیک شبکه نیز افزایش می یابد اما میتوان آن را با مهندسی ترافیک جوری کنترل کرد که با افزایش Throughput ترافیک بسیار کم افزایش پیدا کند .

2) Response Time: **ترجمه زمان پاسخ ، در واقع زمان اجرای یک** Transaction **است و جنس آن زمان است .به عنوان مثال مدت زمانی که کلیک میکنیم تا یک وب پیج باز شود** Response Time **میباشد ولی این** Response Time **تا یک حدی قابل قبول است مثلا اگر این مدت زمان تا 5 ثانیه هم باشد طبیعی است اما اگر دو دقیقه شود قابل قبول نیست و کاربر تا ان زمان صبر نخواهد کرد.**

*** هرچه** throughput **بالا رود** Response time **نیز افزایش می یابد باید تکنیک های کنترل داشته باشیم تا** Transaction **پایین بیاید به جای اینکه بالا رود .**

3) Utilization: **یعنی کارایی در ذهن آن را ترافیک معنی کنید اگر 100% شود شبکه** down **می شود که اصطلاحا میگویند شبکه** bottleneck **شده است . هرچه** throughput **بالا رود ترافیک ،** utilization **هم بالا می رود اما به میزان** Response time **و طراحی سیستم بستگی دارد کارایی یکی از قسمتهای بهره وری است .**

Utilization **کارایی** ~ Traffic Intensity **شدت ترافیک**

مثال u=80% **یعنی 80 درصد آن مشغول است 80 درصد ترافیک دارد**

مثال : 2 تا بانک داریم A,B **برای اینها مدیران** A **و** B **تعریف شده اند هردو بانک 500** throughput**دارند و تعداد کارمندهای هردو 10 نفراست** utilization **بانک** A **30% است اما** utilization**بانک** B **80% است**

پس بانک A **با 3 ساعت کار کردن دارد500 مشتری را راه می اندازد اما بانک** B **با 8 ساعت کار دارد 500 مشتری را راه می اندازد پس بانک** A **موفق تر و بهتر است .**

*****هرچه** Throughput **بالاتر رود** Utilization **نیز افزایش می یابد یعنی این دو با هم رابطه مستقیم دارند .**

نکته : در پایان نامه خیلی مهم است پارامترها رو حتما معلوم کنیم . درفصل 3 راه حل ها باید براساس همین پارامترها باشد وفصل 4 باید نمودارهای این پارامترها رسم شود یعنی اندازه گیری شوند .

نکته : در بحث پارامتر های Performance **در اینجا این سه مورد را انتخاب کردیم و مثال است ولی موارد دیگر را نیز میتوان به عنوان پارامتر معرفی کرد .**

مفهوم *Application Software*:

برای طراحی یک Application **در شبکه مواردی مثل** Throuput **، مدل شبکه و گرافیک آن و کاهش ترافیک شبکه و ... باید درنظر گرفته شود. ابتدا باید ساختار** Application **ای که در شبکه استفاده میشود را بشناسیم دو** Application **تحت**

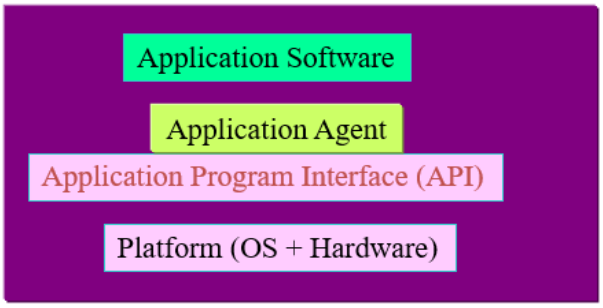
یک API: Application Program Interface و Platform **به هم متصل می گردند**API **به شرکتهای گفته میشود که طراحی** App **میکنند.**

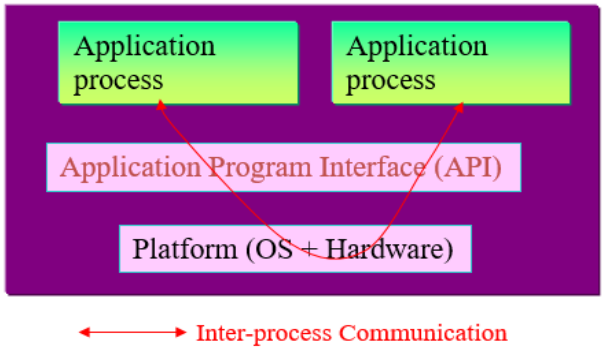
هر softwareApplication **تحت یک مدل شبکه اجرا میشود یعنی هر برنامه ای** platform **اش تحت یک** API **است**

Plat form **: مجموعه ای از سخت افزار و سیستم عاملهای مختلف است که**Application **مورد نظر را باید** support **کنند.**

Application Software

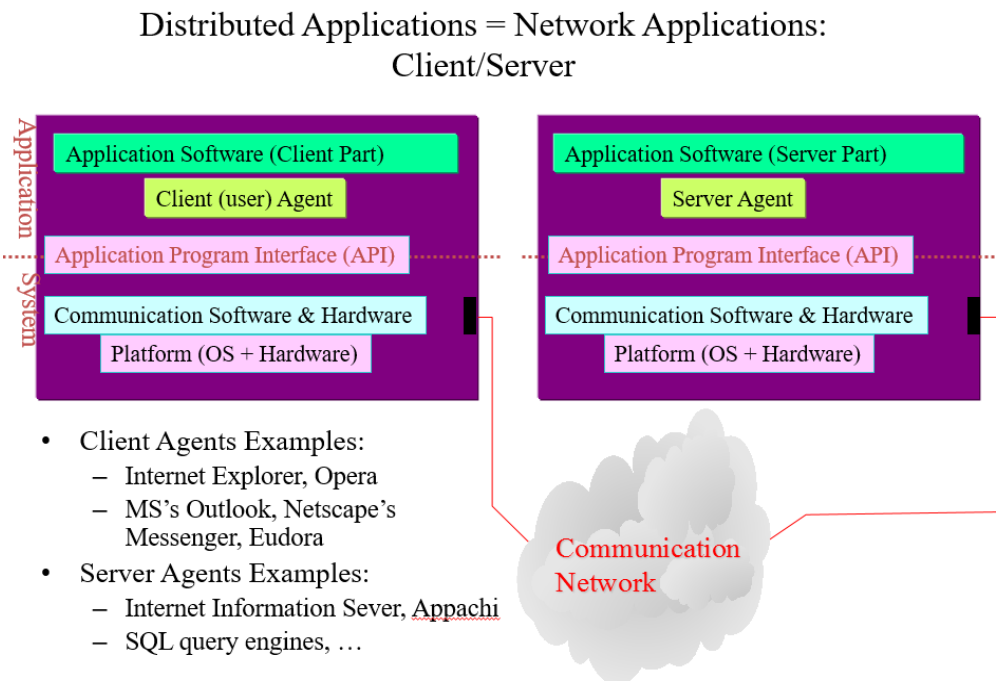
- Platform Services
  - Graphics
  - Data Interchange
  - Data Management
  - User Interface
  - Software Engineering
  - Communication Services





دو اپلیکیشین توسط یک API **و** Platform **خاص با هم** communicate **می شوند .**

جالب این است که در بحث شبکه مدلی هست به نام client/server **که معمولا تمام اپلیکیشن ها وصل میشوند به یک سرور وصل میشوند اکثر** Application **ها به صورت** client-server **کار میکنند .**



Network Application=distributed Application

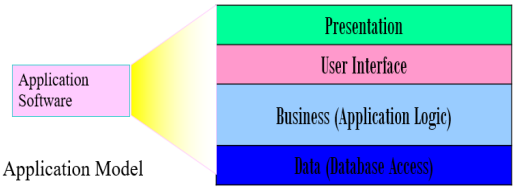
تعداد زیادی Client **وجود دارد که** Application **را استفاده می کنند که به آنها** client agent **می گویند که مثلا برای انجام بازی** clash **و استفاده از** Application **به** Application **سرور وصل می شویم .**

Application Client part **توسط** API **یا** Platform **به** Application Serverpart **وصل میشوند**

دراصل اپلیکشین ها وقتی بحث میشود 2 بخش داریم بحث اپلیکیشنی وبحث سیستمی شبکه ای آن اپلیکیشن داریم بحث اپلکیشنی لایه یک بحث سیستمی 2 لایه دیگر است

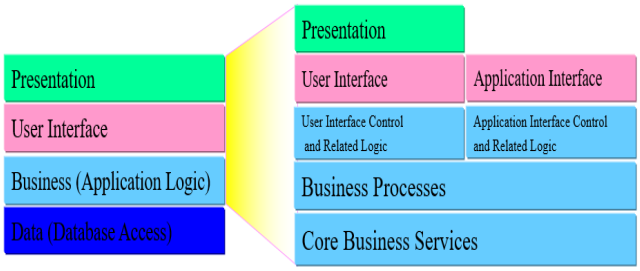
بعدا خواهیم خواند که ما بعضی وقتی 2 لایه و بعضی وقتها 3 لایه comminucate **میکنیم .**

Layered Application Model-1



- Application Model
  - Presentation: The client agent remains focused on presenting information to or receiving input from the user.
  - User Interface: User's access to the application logic via client agent. It can be dynamical and configured by user. It is build on the top of the user interface control.
  - Dynamic User Interface:
    - Customizing the look (example: www.cstore.com
    - Customizing the content (examples: my.yahoo.com , www.exite.com)

Layered Application Model-2



- Business Rules (Application Logic)
  - Units of processing or algorithms that represents concept of importance to the organization using database.
- Data (Database Access)
  - Logic to connect to database; access/manipulate data held within databases.

***2 لایه** Application **و** System **جدا از هم نیست .**

مدل لایه ای Application **:**

مدل سه لایه : Tier client/server **- 3به** Application Network **هایی که تحت سه لایه طراحی می شوند گفته می شودلایه**

اول : Presentation **یا** User Interface**لایه دوم**: Business**لایه سوم** : Data

"Logical Tiers vs Physical Tiers

- Application Model
  - Logical Tiers
    - Presentation
    - User Interface
    - Business
    - Data
  - Physical Tiers
    - Client workstation
    - Application server
    - Data Base

Presentation User Interface	Client Workstation
Business (Application Logic)	Application Server
Data (Database Access)	Database

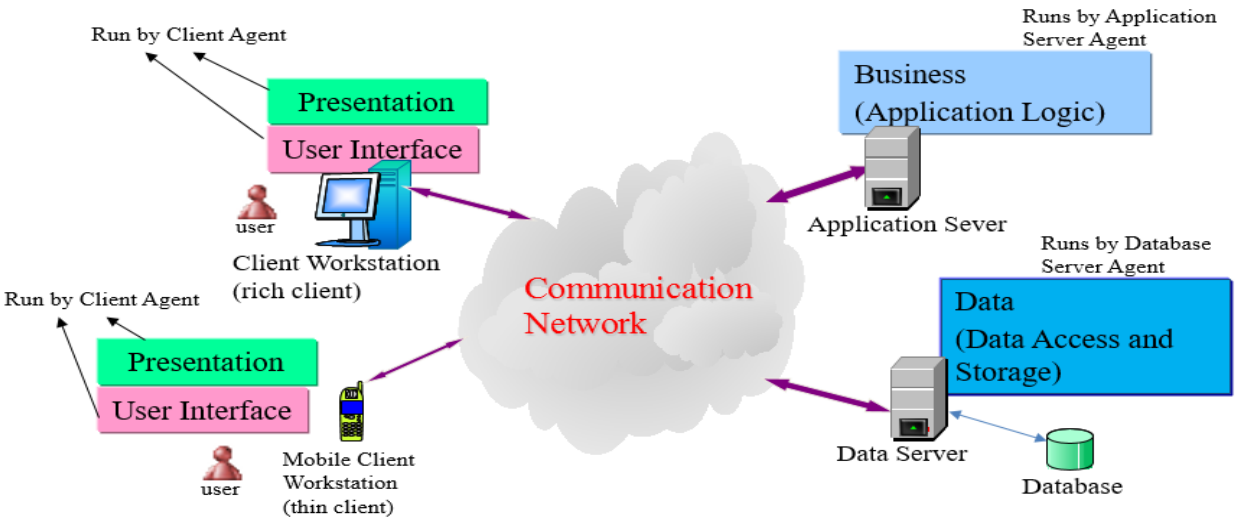
مثالی که برای لایه یک است ، Google **می باشد که یک صفحه با یک** Textbox **در وسط دارد که بالاترین سطح** presentation **است و هرکس می تواند توسط** Textbox **جستجو کند و نیاز به آموزش ندارد که اصطلاحا** User interface **نام دارد.**

در شکل زیر Clientworkstation **مثل دفتر هواپیمایی است که تعداد زیادی وجود دارد** Application **ای که روی کامپیوتر آنها نصب است جز لایه یک است . لایه یک فقط مخصوص ارسال** request **است مثلا درخواست بلیط بلیط می دهیم که یک** request **است که این درخواست به لایه 2 ارسال میگردد. لایه 2** Application Server **است .**

Application Server **همه درخواستها را** process **میکند.**

Request **از سمت** workstation **در لایه 1 ارسال میشود به** Application Server **که محل** Run **شدن** Request **است**

Data Server **که شامل** Database **است در لایه 3** communicate **وجود دارد. تا زمانی که** process **در** Application Server **کامل شود پس از کامل شدن** Reply **به لایه 1 (**presentation**) ارسال می گردد.**



معمولا Database Server **در جایی خارج از سازمان است تا امنیت داشته باشد و اطلاعات از بین نرود.**

لایه 1 لایه presentation **است یا** user interface **این لایه اسمش روش است خودش خودش را معرفی میکند و ظاهر آن کاربر پسند باشد.**Google **بالاترین سطح** presentation **دارد و هرکاربری راحت با ان کار میکند باید** user interface **بودن خیلی مهم است .توسط** client agent **این قسمت** run **می شود**

محل لایه 1، Client workstation (rich client)**روی کامپیوتر است** .

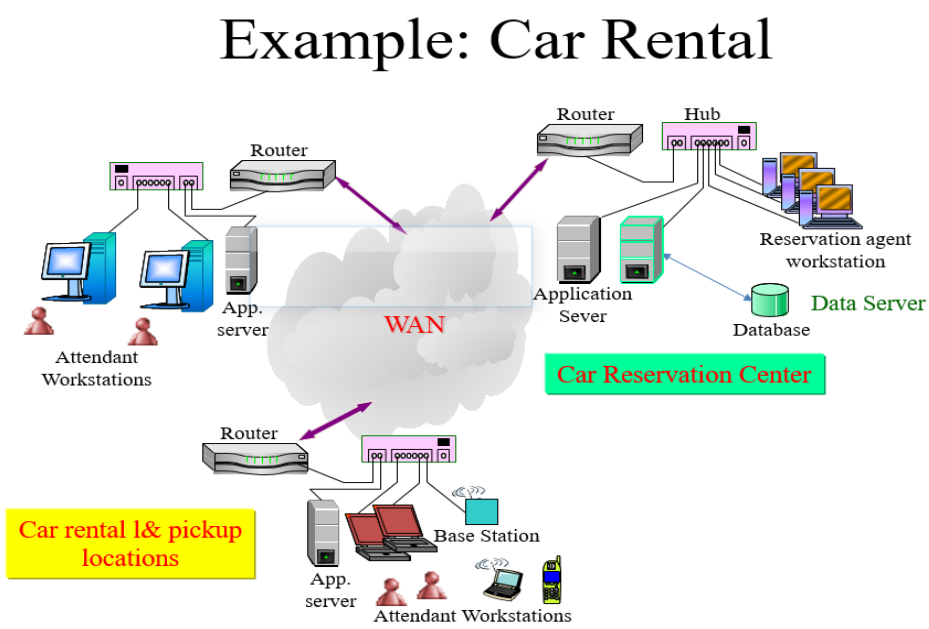
و Mobile client workstation (thin client) **روی موبایل تبلت است** .

Client workstation presentation **پیاده سازی می شود .**

لایه 2 : لایه ی business **است که مربوط به پردازش اطلاعات است** Application **در این لایه قرار دارد لایه ی** Business **روی** Application Server **پیاده سازی می گردد.**

لایه 3: لایه ی Data **است که اطلاعات در این لایه قرار می گیرد لایه** Data **روی** Database **پیاده سازی می شود.**

برای یادگیری راحت ، وقتی یک قرمه سبزی قرار است درست شود از طرف شخصی درخواست داده میشود از لایه 1 درخواست داده شده است آشپزخانه میشود Application server **که محل اجرا است لایه 2 بعد باید مواد اولیه را تهیه کرد با زنبیل قرمز میرید به لایه 3 یعنی** Database Server **یعنی دیتابیس سرورهای مختلف مراجعه میکند تا مواد اولیه را تهیه کند یعنی بارها باید به دیتابیس های مختلف مراجعه کند .**



Increasing the load

یک رابطه بین throughput و Response time **وجود دارد در شبکه برای کارکردن با یک** Application **بایدتعداد** Transaction **ها مشخص باشد درجدول بعد 4 تا** Transaction **وجود دارد اولین قدم در طراحی هر نرم افزار مشخص کردن تعداد** Transaction**ها است.**

هر Transaction **دارای** Throughput **اولیه است**

درجدول زیر با 100=throughput **برای** local reservation =transaction **،** responcetime **برابر** 1/28 **است.**

اگر throughput **به اندازه 5% افزایش یابد** Responcetime **برابر 1/67می گردد.**

اگر throughput **به اندازه 10% افزایش یابد** Responcetime **برابر 2/45می گردد.یعنی 2 برابر می شود**

اگر throughput **به اندازه 15% افزایش یابد** Responcetime **تقریبا 4 برابرمی گردد.**

هرچه قدر throughput **افزایش یابد ،** Response time **نیز زیاد می گردد.**

پس در طراحی نرم افزار تعداد Transaction **ها واینکه هر** Transaction **چه** Throughput **و** Responcetime **ای دارد باید مشخص باشد .**

Increasing the Load

- The load (transactions)
  - Local reservation
  - (at car rental location)
  - Road assistance request
  - Car pickup
  - Phone reservation

Transaction	Current Load Intensity	CLI +5%	CLI +10%	CLI +15%
Local Res.	1.28	1.67	2.45	5.06
Road Ass.	0.64	0.87	1.37	3.20
Car Pickup	0.64	0.76	0.94	1.23
Phone Res.	0.85	1.16	1.82	4.24

Table 1.2. Response Time for Various Load Values (sec)

در شبکهcar rental **قبل** Application Server **و** Data Server **دو کامپیوتر مختلفند ولی در یک شبکه هستند بعضی اوقات** throughput **انقدر بالا میرود که شبکه** Down **می شود.بنابراین باید تکنیک های به کار ببریم که از این اتفاق جلوگیری کنیم . یکی از این روشها استفاده از** Cache **است** .

*** *اینترنت چیست ؟ به مجموعه ای از اتصال سرورها گفته میشود (در واقع به دسترسی به چند سرور اینترنت میگویند)به عنوان مثال وقتی کلمه ای را در گوگل تایپ میکنیم و سرچ را میزنیم نتایج حاصل هر کدام از یک سرور و از یک کشور هستند بنابر این ما به چند سرور دسترسی پیدا کردیم و این مفهوم ساده اینترنت است .**

در شکل زیر پهنای باند شبکه پایینی Mbps**ها است وتوسط** Router **به اینترنت وصل است و پهنای باند** link **ارتباط آن به اینترنت** 1.5mbps **است فرض کنید معدل اندازه یک بسته اطلاعاتی که در این شبکه کار می کند برابر با** L=100000bit **است.**

از نرم افزارهای benchmark **می توان استفاده کرد تاشبکه را** monitor **کند و اندازه** Packet **را به دست بیاورید که معدل** Packet**ها را اعلام میکند .**

a **= قدرت سرور است که در شبکه زیر** req/sec **15 است "سروری که در شبکه** LAN **است می تواند 15 درخواست را در هر ثانیه** Support **کند"(توانایی سرور است)**

R **: پهنای باند**

$$Utilization\ or\ Traffic\ Intensity\ on\ LAN = \frac{\text{a قدرت سرور} \times \text{L طول بسته}}{\text{پهنای باند } R}$$

Caching example 1

فرضیات

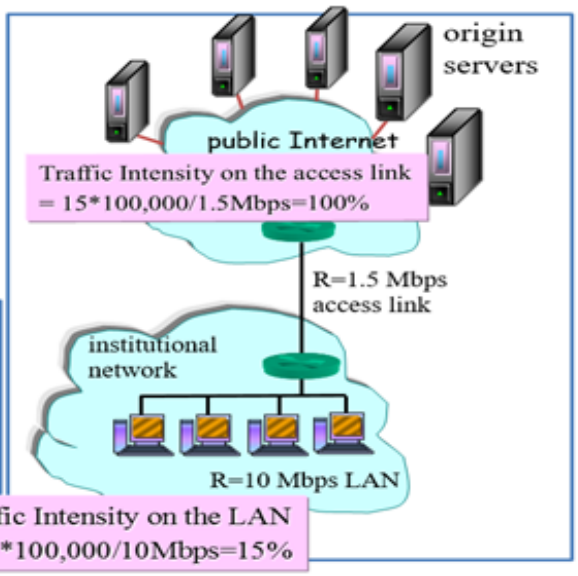
Assumptions

- average object size = L= 100,000 bits
- avg. request rate from institution's browser to origin serves = a= 15reqs/sec
- delay from Internet router to any origin server and back to router = 2 sec

Consequences

نتایج

- utilization on LAN = 15%
- utilization on access link=100%
- total delay = Internet delay + access delay + LAN delay
  = 2 secs + minutes + 20 msecs



تمام ترافیک 15% **روی لینک است پس ترافیک روی** Link **هم وجود دارد.**

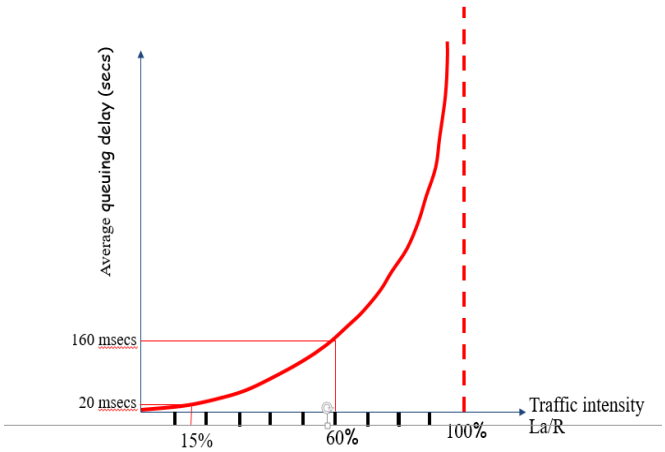
$$Traffic\ Intensity\ on\ LAN = \frac{a \times L}{R} = \frac{a \times L}{R \times 10^6} = \frac{15 \times 100000}{10 \times 10^6} = \frac{15}{100} = 15\%$$

15% ترافیکی عالی محسوب میشود یعنی 15 درصد ترافیک داریم پهنای باند اشغال است .

(برای تبدیل mb **به** bit **در** 10^6 **ضرب میکنیم تا صورت و مخرج از جنس بیت باشند)**

$$traffic\ Intensity\ on\ the\ link = \frac{a \times L}{R \times 10^6} = \frac{15 \times 100000}{1.5 \times 10^6} = \frac{15}{15} = 100\%$$

چون در شبکه ترافیک داریم پس مشکلی وجود ندارد ولی چون ترافیک لینک 100% **است پس شبکه** Down **است،** Bottleneck **است .**



برای حل مشکل فوق راههای زیر وجود دارد : فاز finding problem **مهم ترین فاز است :**

1)پهنای باند را زیاد میکنیم 1.5 ⟵ **۱۰ افزایش یا کاهش پهنای باند لینک روی شبکه تاثیری ندارد پس**

Traffic Intensity**از 100% به 15% می رسد ، ترافیک لینک از ترافیک شبکه مستقل است .**

برای اینکار Link **راعوض کنیم که ممکن است هزینه ی زیادی مصرف کنید که مدل** cost **این راه را رد میکند حرف اول و آخر را مدل** Cost **میزند باید راه حلی انتخاب گردد که مدل** Cost **آنرا تائید کند.**

به عنوان مثال برای کسی که سردرد دارد دو نوع نسخه میتوان داد ،یک اینکه برای آن شخص یک سفر خارج چند روزه تجویز کنیم و حالت دوم دوم اینکه یک عدد قرص مسکن برایش تجویز کنیم ، خروجی هر دو یکی است و مساله را حل میکنند اما مساله cost model **در این جا به وجود می آید.**

Caching example 2

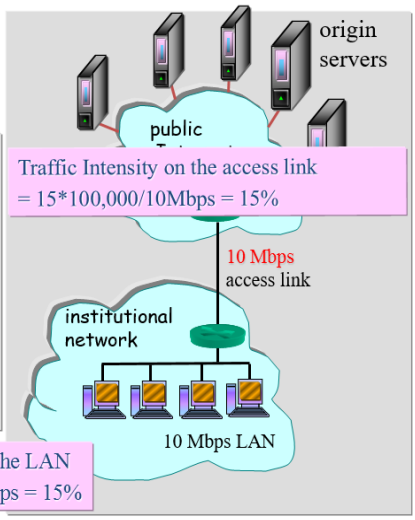
Possible solution

- increase bandwidth of access link to, say, 10 Mbps

Consequences

- utilization on LAN = 15%
- utilization on access link = 15%
- total delay = Internet delay + access delay + LAN delay

  = 2secs + 20msecs + 20msecs
- often a costly upgrade



4-11

نکته) برای اینکه بفهمیم متوسط اندازه Packet **یک شبکه چقدر است باید نرم افزار** benchmark **را به شبکه وصل کرد کار این نرم افزار مانیتور کردن شبکه است .**

2)یکی دیگر از راه حل ها استفاده از مدل Cache **است در واقع** install **کردن** Cache Server **در شبکه** LAN **که داری 2 نرخ زیر است**

- *Hit rate*
- *Miss rate*

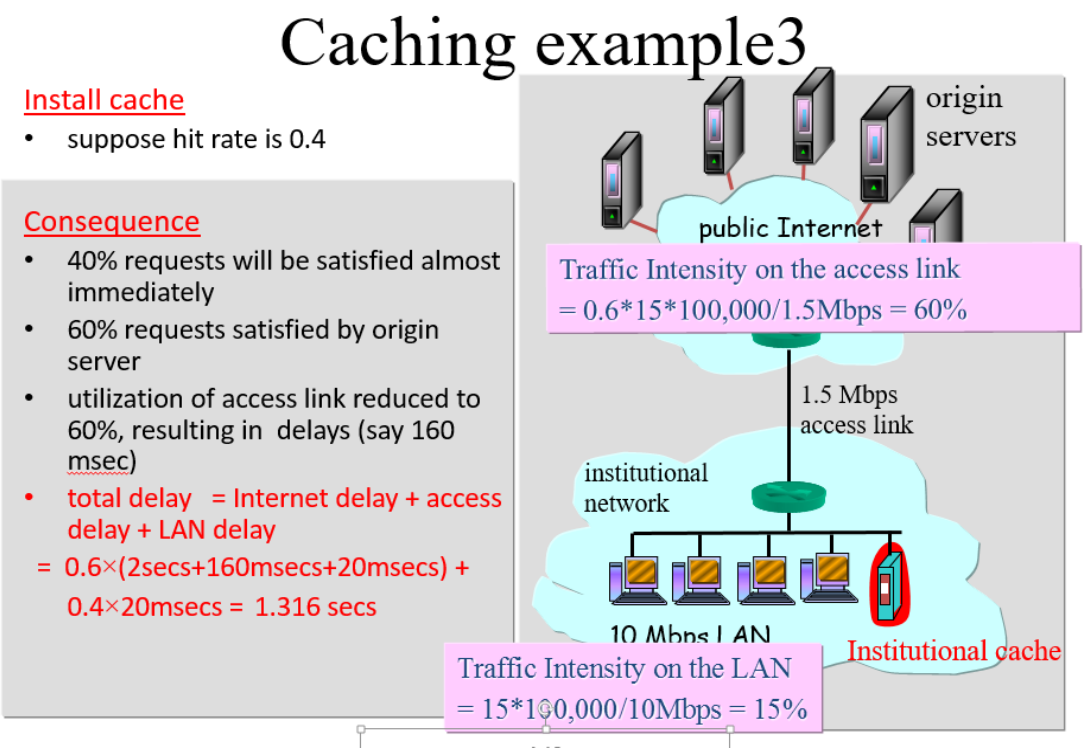
Cache **چیست ؟حافظه ای سریع داخل** CPU **است حافظه ای کوچک است اماخیلی با ارزش می باشد که به دلیل** hit rate **با ارزش شده است .**

اگر Request **به** cache **ارسال شود و پاسخ** Request **در** cache **باشد مکانیزم** hit rate **است ولی اگر پاسخ** request **در** cache **نباشد مکانیزم** miss rate **است .اگر** hit rate **برابر 0.4 باشد یعنی ازهر 10تا** Request **، 4 تای آن پاسخ پاسخ دارد و 6 تا هم پاسخ ندارد پس** miss rate **برابر 0.6 است** hit rate, Miss rate **مقابل هم هستند .**

اگر cache **ی با** hit rate=0.4 **را در شبکه** LAN **نصب کنیم** client **ها ابتدا** request **را در** cache **می بیند اگر پاسخ** request **در** cache **باشد** request **را روی لینک نمی فرستند**

به اندازه hit rate **،** cache **نرخ ترافیک لینک کاهش می یابد**

پس به جای تعویض لینک که هزینه بر است از cache server **در شبکه استفاده میکنیم**



درصورتی که Cache **سرور در شبکه نباشد همه** 15req/s **روی لینک فرستاده میشود و** missrate **می گردد اما درحالتی که** cache **سرور وجود داشته باشد فقط 0.6 درخواستها روی لینک فرستاده میشود و 0.4 آن روی** Cache **بوده است**

$$traffic\ Intensity\ on\ the\ link = \frac{a \times L}{R \times 10^6} = \frac{0.6 \times 15 \times 100000}{1.5 \times 10^6} = \frac{0.6 \times 15}{15} = 60\%$$

میبینم که ترافیک از 100% به 60% رسید ترافیک 60 ترافیک روان و خوبی است .

تمرین :

درشکل زیر کاربران 2 LAN **فایل هایی مشخصات زیر را از سرورها درخواست می نمایند متوسط زمان دریافت یک فایل** L1 **بیتی را توسط کاربران** LAN1 **حساب کنید هر دو ازیک** Cache server **استفاده میکنند .**

$L_{1=100000\ bits}$ ، $a_{1=15\ req/sec}$ ، $h_1=50\%\ (hite\ rate)$

$L_{2=50000\ bits}$ ، $a_{2=20\ req/sec}$ ، $h_2=30\%\ (hite\ rate)$

$Internet\ delay = 2secs$

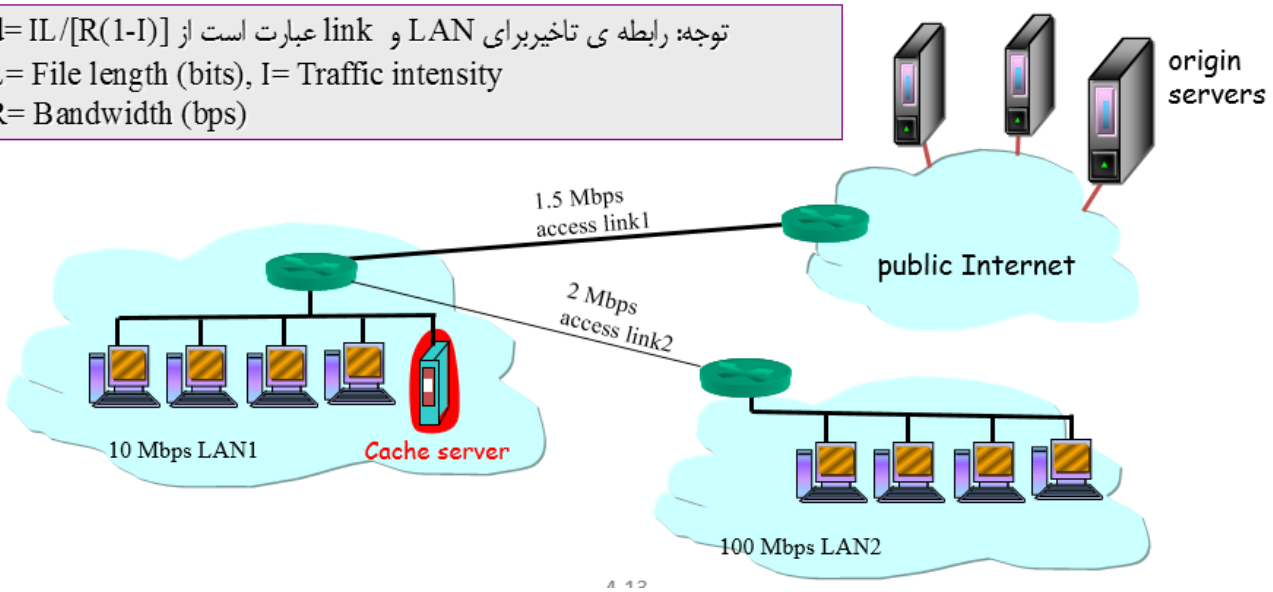
Traffic Intensity on LAN1=?

Traffic Intensity on LAN2=?

Traffic Intensity on link1=?

Traffic Intensity on link2=?

توجه: رابطه ی تاخیربرای LAN و link عبارت است از d= IL/[R(1-I)]
L= File length (bits), I= Traffic intensity
R= Bandwidth (bps)



تأخیر *Communication-Processing Delay*

تأخیر در ارسال و دریافت Packet **مورد بحث است هر** Resource **یک** Response time **دارد در مدل 2 لایه ای زیر**

Waiting time: **مدت زمانی که** Request **در** Queqe **صف است .**

Service time: **مدت زمانی که سیستم درحال سرویس دادن به** Request **است .**

Residence time: Waiting time + Service time **که همان** Responcetime **است اما برای یک** Resource .

Response time : **مجموعه** Residence time **است یا زمان اجرای یک** Transaction .

Communication-Processing Delay

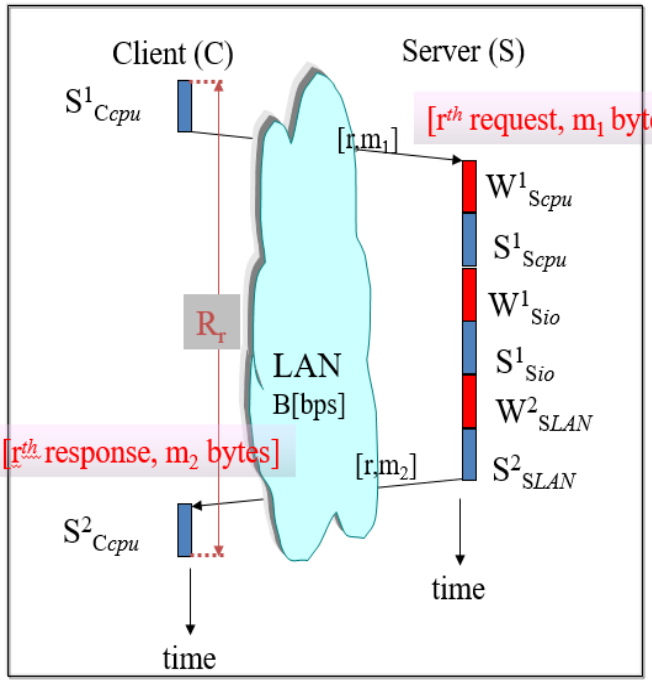
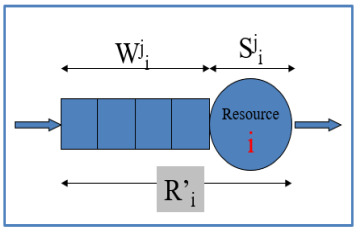


Figure 3.1. Communication-Processing delay diagram for 2-tier C/S system.

The Times

- Service Time during j^{th} visit: S^j_i
- Waiting Time during j^{th} visit: W^j_i
- Residence Time during j^{th} visit: R'_i



$$R_r = R'_{Ccpu} + R'_{Scpu} + R'_{Sio} + R'_{LAN} \quad (3.2.5)$$

R_r = response time of a request r

R'_{Ccpu} = residence times at the client cpu

R'_{Scpu} = residence times at the server cpu

R'_{Sio} = residence times at the server io

R'_{LAN} = residence times at the LAN

Residence Time

Service Demand (sum of all service time for a request at source i): D_i

$D_i = \Sigma_j S^j_i$ (3.2.2)

Queuing Time (sum of all waiting time for a request at source i): Q_i

$Q_i = \Sigma_j W^j_i$ (3.2.3)

$R'_i = D_i + Q_i$ (3.2.4)

$R_r = \Sigma_i R'_i$ (3.2.5)

$D_{Scpu} = S^1_{Scpu} + S^2_{Scpu}$

Service demand at server's *cpu*

$D_{LAN} = 8(m_1 + m_2)/B$

m_1 = request length [B]

m_2 = reply length [B]

$Q_{Scpu} = W^1_{Scpu} + W^2_{Scpu}$

Queuing time at server's *cpu*

مجموع زمان تمام سرویس های شکل قبل = Service Demand

مجموع زمان تمام انتظارها=Quening time

$D_i + Q_i$ = Residence time پس

Example 3.1

S^1_{Ccpu} = 5ms, S^1_{Scpu} = 10ms

AvgSeek = 9ms (Server's Disk)

AvgLatency = 4.17msec (Server's Disk)

TransferRate = 20 MB/s (Server's Disk)

DataReads = 10×2048 Byte

S_d = 9+4.17+2048/20M = 13.3 ms

(Disk Average Service Time)

$D_d = 10 \times S_d$ = 133 ms

(Service Demand at Server's Disk)

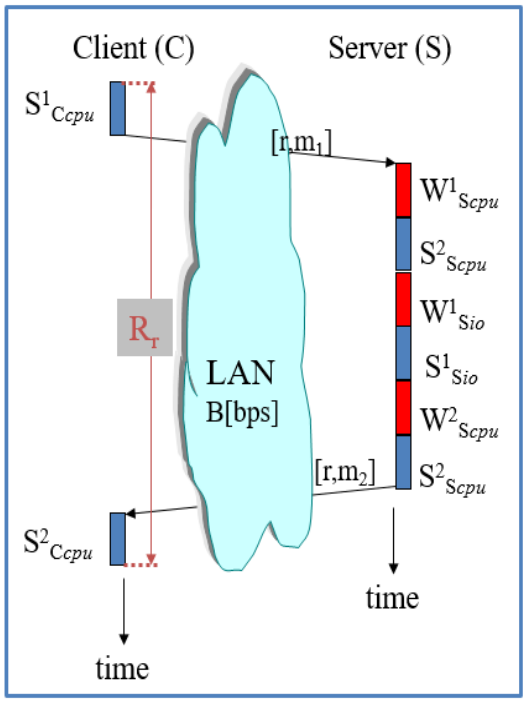
m_1 = 1,518, m_2 = 7×1,518 Bytes

$D_{LAN} = 8(m_1 + m_2)/10Mbps$= 9.7ms

$R_t > D_{Ccpu} + D_{Scpu} + D_d + D_{LAN}$=158 ms

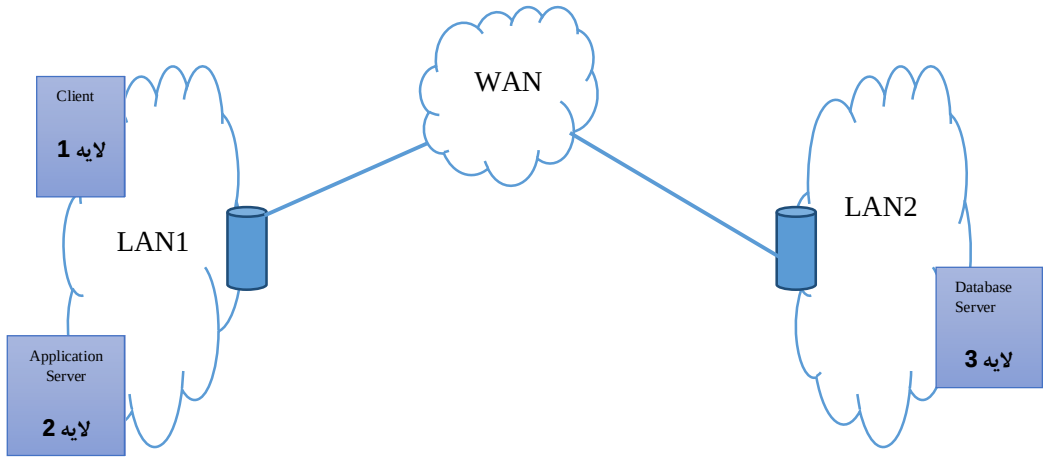
The minimum value for response time

to transaction t; all waiting times are ignored.



اگر بسته m_1 بایت ارسال گردد چون پروسه روی آن انجام می شود m_2 بایت برمی گردد.

ولی در مدل 3 لایه زیر داریم : Client **در لایه یک و** Server **در لایه 2(لایه تجاری) است .که لایه 1 و 2 در یک** LAN**هستند و لایه 3 در** LAN **دیگری است که توسط** WAN **به هم مرتبط هستند .** LAN **ها در کشورهای مختلفی هستند.**



تأخیر بسته ای که از لایه 1 می آید به چه صورت است ؟

بسته از Client وارد LAN1 میشود و به Application Server می رود مجددا وارد LAN1 شده و WAN می رود سپس وارد LAN2 می گردد و به Database Server می رود . بین Application Server , Database Serverبارها رفت و آمد می کند. Response time , Request timeمجموع تمام عملیات فوق است .

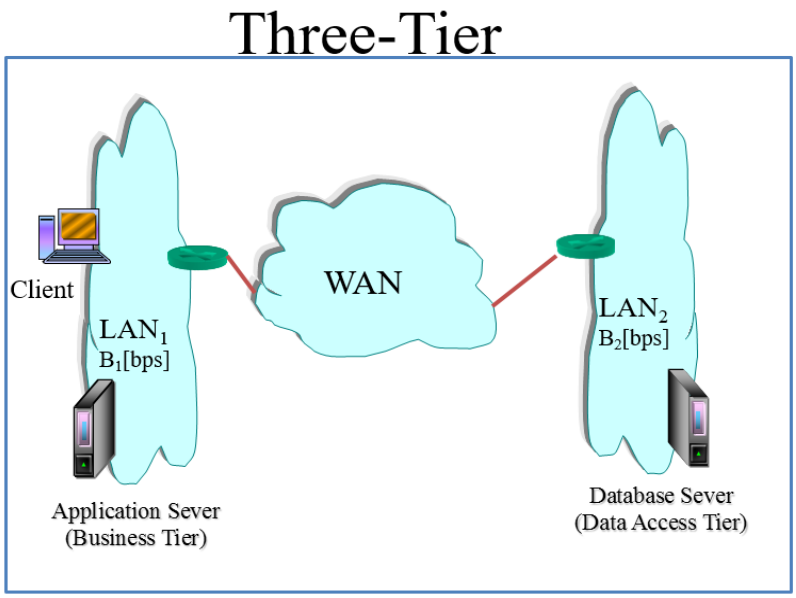


Figure 3.3. Three-tier C/S system.

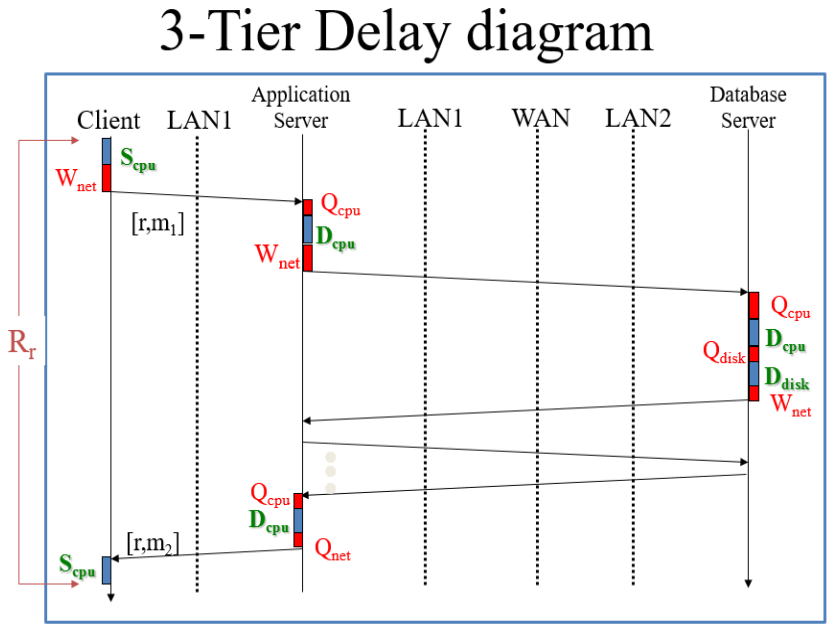


Figure 3.4. Communications-processing delay diagram for a 3-tier C/S system.

چون بسته بارها بین Database Server **و** Application Server **رفت و امد می کند پس به جای** S,W **و** Q,D **درنظر میگیریم** Qdisk **مربوط به بافر دیسک سرور است . دیسک نیز** Service dimand **دارد یکی از مشکلات سیستم عامل این است که سرعت** CPU **از سرعت** disk **بالاتر است پس** Os **از مکانیزم** Interup **استفاده میکند تا سرعت** hard disk **به سرعت** cpu **برسد (درواقع یک وقفه می دهد)**

پس R_r **مجموع تمام زمانهای فوق است .**

سوال : چرا در Application Server **در قسمت بلا** Wnet **است امادر پایین** Qnet **است ؟**

چون کلاینت یک درخواست ارسال میکند و بارها process **اتفاق می افتد برای آن یک درخواست**

Validating :

در IT **و در پایان نامه نمونه گیری انجام شده باید تعدادی باشد که** valid **و معتبر باشد** Validating **زمانی درنظر گرفته می شود که بخواهیم مدلمان با یک مقدار واقعی مقایسه گردد .**

فصل چهارم

شبیه ساز شبکه NS2

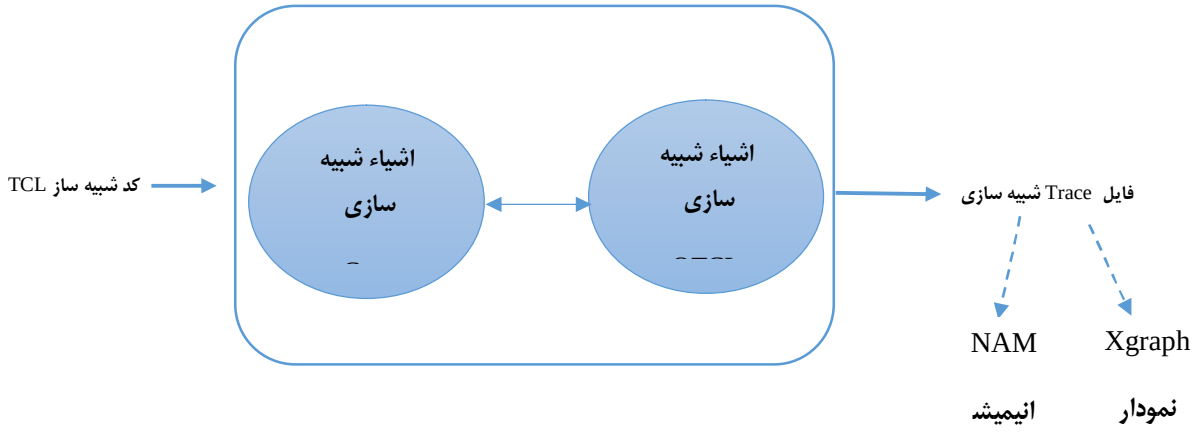
مهد Network در دانشگاه کالیفرنیا است بنیاد ملی ایالت متحده آمریکا یکی از اسپانسرهایش بوده و ISI Information Science Institute از اسپانسرهای NS2 است .

معماری و سختار NS2

یکی از مزایای NS2 ، open source **بودن آن است** Ns2 **از 2 زبان برنامه نویسی استفاده میکند** (C++, Otcl) **که همزمان اجرا شده تا بتوانند یک کد شبیه سازی را** Run **کنند .**

در شبیه سازی با دو پارامتر مواجه هستیم :

- ✓ **سرعت اجرا : سرعت اجرای** C++ **بسیار بالا است ولی برای** OTCL **بسیار پایین است**
- ✓ **سرعت تغییرات : سرعت تغییرات** C++ **بسیار پایین است ولی برای** OTCL **بسیار بالا است**



دستور قابل اجرای NS2 یعنیNS

معماری کلی NS2

در شکل فوق که TCL **وارد شبیه ساز شده و خروجی آن یک فایل** trace **است که با 2 حالت خروجی را نمایش می دهد .**

NAM **: نرم افزار انیمیشن است که به صورت گرافیکی نمایش میدهد**

Xgraph **: که خروجی را به صورت نمودار نمایش میدهد که در** Science **از** Xgraph **استفاده نمیکنیم چون برای یک نمونه آزمایش استفاده می گردد.**

رابطه بین OTCL **و** C++ **در کلاس ها می باشد یک کلاس از** OTCL **و یک کلاس از** C++ **باهم سنکرون می شوند و ارتباط برقرار میکنند به همین دلیل گفته می شود که** NS2 **پیچیده است .**

کاربرد NS2 **: برای شبیه سازی شبکه های با سیم و بی سیم / ماهواره ای / مسیریابی/ انتقال** TCP , UDP

MultiCast ,UniCast **/ منابع ترافیکی** (Web,Ftp,Telnet)

مزایا NS2

1. **از 2 زبان استفاده میکند**
2. **خیلی از پروتکل ها در** Ns2 **پیاده سازی شدند و فقط فراخوانی می شوند**
3. **خیلی از مدل ها را می توان با** NS2 **ساخت**
4. Open source **است**

معایب NS2

مدت زمان طولانی برای آشنایی با آن

زبان TCL **دستورات**

درج توضیحات : با علامت #**مینویسند ، هرچه درجلوی** # **نوشته شود در خروجی برنامه نمی آید ولی در** Source code **دیده میشود چون تعداد خطوط بالا است** comment **نوشتن بسیار کاربردی است مثلا میتوان با** Comment **ابتدا و انتهای کد را مشخص کرد**

So we are to sratt programing with TCL

اعلان متغییر :برای تعریف متغییر از کلمه SET **استفاده میکنیم دستور زیر یعنی 12 را در** a **قراربده**

set a=12

زبان TCl **همه چیز را رشته می بیند مثلا** 1+1 **از نظر عددی یعنی 2 ولی از نظر رشته 11 است یعنی کاراکتر را کنار هم قرار میدهد**

کلمه Hello **را داخل متغییر** b **قرار میدهد** Set b hello

اگر 2 کلمه را بخواهیم در متغییر قرا ردهیم باید داخل " " بنویسیم Set c “hello world”

یک کلمه ای را هم میتوان داخل " " نوشت Set b “hello”

چون بین کلمات فاصله نیست میتوان از " " استفاده نکرد Set c helloworld

اگر بخواهیم متغییری که استفاده کردیم را حذف کنیم از unset **استفاده میکنیم** Unset c

یعنی محتوای a **را در** d **قراربده** Set d $ a

یعنی محتوای متغییر را چاپ کن Puts $a

510 چاپ میکند (چون رشته است) Puts 5+10

15 چاپ میکند Puts [expr 5+10]

تابع expressional **کمک میکند که از رشته به عدد تبدیل شوند**

مثال: برنامه ای که عدد3 را در متغییر ش و عدد 7 را در متغییر b **قرار داده این دو متغییر را درهم ضرب کرده و حاصل را در متغییر** c **قرار داده و نهایتا مقدار** c **را در خروجی چاپ کند**

```
#define a=3 and b=7
Set a=3
Set b=7
# now define c=a*b
Set c [exp {$a*$b}]
Puts “c is $c”
```

خروجی c is 21

مقدمه

شبیه ساز شبکه (نسخه ۲) که با نام NS2 شناخته شده است، بطور ساده یک ابزار شبیه سازی رخداد-گسسته است که در مطالعه طبیعت دینامیکی شبکه های ارتباطی، مورد استفاده قرار میگیرد. به کمک NS2 شبیه سازی پروتکلها و عملکردها (مثل TCP و UDP و الگوریتمهای مسیریابی و ...)ی شبکه های سیمی و شبکه های بی سیم بخوبی قابل انجام هستند. همچنین امکان تغییر و تعریف پروتکلهای شبکه ای و رفتار متناظر آنها توسط کاربران فراهم شده است.

انعطاف پذیری و طبیعت ماژولار این شبیه ساز، سبب محبوبیت و استفاده همیشگی آن در مجامع تحقیقاتی شبکه (از زمان ایجاد آن در سال ۱۹۸۹) شده است. از آن موقع این ابزار دستخوش تغییرات و تجدید نظرهای سازنده ای شده است که دست اندرکاران این زمینه سهم قابل توجهی در آن داشته اند. از جمله این دست اندرکاران میتوان به موارد زیر اشاره کرد:

- دانشگاه کالیفرنیا [۱] و دانشگاه کورنل [۲] که شبیه ساز شبکه **واقعی** [۳] را ایجاد کردند، و این شبیه ساز ، سنگ بنای NS محسوب میشود.
- آژانس DARPA [۴]، که از سال ۱۹۹۵، و از طریق پروژه VINT [۵] ، توسعه NS را پشتیبانی کرد.
- بنیاد ملی علوم [۶] (ایالات متحده) ، که اخیرا به جمع توسعه دهندگان NS پیوسته است.
- ISI [۷]، که در حال حاضر توسعه NS2، توسط این انجمن علمی صورت میگیرد.
- و در آخر (و نه به لحاظ درجه اهمیت)، محققان و توسعه دهندگانی که در مجامع مختلف با فعالیتهای مداوم خود، NS2 را پر قدرت و فراگیر نموده اند.

همانطور که گفته شد، درحال حاضر توسعه NS2 بوسیله ی ISI انجام می شود و توسط DARPA و NSF پشتیبانی می شود.

NS2 اساسا برای کار در محیط لینوکس ساخته شده است، اما میتوان به کمک برنامه های کمکی مثل cygwin [۸](که یک محیط مجازی از سیسم عامل لینوکس ایجاد میکند) آن را روی سیستم عامل windows نیز نصب نمود. جزئیات نصب در ویندوز، بصورت یک فایل ویدیویی تهیه شده است.

۱- University of California

۲- Cornell University

۳- REAL Network Simulator: در اصل به عنوان ابزاری برای مطالعه رفتار دینامیکی جریان و طرحهای کنترل تراکم (congestion) در شبکه های داده ای سوئیچ-بسته (packet-switched data networks)، پیاده سازی شده بود.

۴- Defense Advanced Research Projects Agency

۵- Virtual InterNetwork Testbed : توسط آژانس DARPA و با هدف ایجاد یک شبیه ساز شبکه که بتواند مطالعاتی در زمینه پروتکلهای مختلف و شبکه های ارتباطی انجام دهد، ایجاد شد.

۶- NSF: National Science Foundation

۷- Information Science Institute

۸- یک محیط شبه لینوکسی (Linux-like) برای ویندوز است و شامل دو قسمت است، یک قسمت، DLL که مثل API لینوکس عمل میکند و قسمت دیگر مجموعه ای از ابزار برای فراهم کردن ظاهر و کارایی لینوکس. این نرم افزار محصول شرکت red hat و با شعار cygwin=Cygnus+GNU+windows میباشد. [مرجع: http://cygwin.com]

معماری و ساختار NS2

NS2 بر پایه دو زبان بنا شده است: یکی شبیه ساز شی گرا که به زبان C++ نوشته شده و دیگری مفسر [۱] OTcl [۲] که برای اجرا کردن دستورات برنامه های کاربران استفاده می شود. C++، مکانیسم داخلی اشیاء شبیه سازی را تعریف میکند (در پشت صحنه!) و OTCL بوسیله اسمبل و پیکربندی اشیاء، شبیه سازی را برپا میسازد (روی صحنه!).

تذکر: قطعات NS2 عبارتند از خود شبیه ساز NS و NAM [۳] که انیماتور شبکه بوده و باعث بصری سازی [۴] خروجی ns می شود.

حال ممکن است این سوال مطرح شود که چرا NS از دو زبان برای شبیه سازی استفاده میکند؟ در پاسخ به این سوال باید گفت:

از یک طرف، در شبیه سازی هایی که در آنها پروتکلها با جزئیات زیاد پیاده سازی میشوند، زبان برنامه نویسی ای لازم است که بتواند بایتها، بسته ها و سرآیندها را بطور مطلوب دستکاری کند و الگوریتمهایی را پیاده سازی کند که قرار است روی حجم زیادی از داده ها اجرا شوند. در این حالت سرعت زمان اجرا اهمیت بیشتری نسبت به زمان پروسه تغییر و اجرای مجدد (که شامل اجرای شبیه سازی، یافتن خطاها، رفع خطاها، کامپایل مجدد و اجرای مجدد است،) دارد.

از طرف دیگر، حجم عمده ای از تحقیقات شبکه ای، شمار اندکی از پارامترها یا پیکربندیها را در بر دارند و به سرعت چند سناریو را کاوش میکنند. در این حالت زمان تکرار مجدد (یعنی تغییر مدل و اجرای دوباره آن) بسیار مهمتر است. از آنجایی که در این حالت، پیکربندی اولیه فقط یکبار (و در زمان آغاز شبیه سازی) اجرا میشود، زمان اجرا اهمیت کمتری خواهد داشت.

NS هر دوی این نیازها را به کمک دو زبان برنامه نویسی برآورده میکند. C++ زمان اجرای بالایی دارد حال آنکه در اعمال تغییرات کند است؛ بنابراین برای پیاده سازی پروتکلهای با جزئیات زیاد مناسب است. در مقابل، OTCL بسیار کندتر اجرا میشود اما در عوض در اعمال تغییرات بسیار سریع است و بصورت تعاملی عمل میکند؛ بنابراین برای پیکربندی شبیه سازی کارآیی ایده آلی دارد.

همانطور که در شکل ۱-۱ ملاحظه میشود، کد شبیه سازی که به زبان TCL نوشته میشود به کمک دستور ns (دستور قابل اجرای NS2) اجرا میشود. خروجی، فایل Trace (با فرمت tr.) است که در بیشتر موارد از آن برای رسم نمودار یا متحرک سازی (انیمیشن) شبیه سازی استفاده میشود. در مورد تمامی این موارد در قسمتهای بعدی توضیحات کاملتری ارائه خواهد شد.

C++ و OTCL با استفاده از TclCL به هم پیوند میخورند. یعنی در واقع TclCL (TCL به همراه کلاسهایش[۵]) واسط میان C++ و OTCL است؛ به عبارت دیگر، امکان استفاده اشیاء و متغیرها را در هر دو زبان فراهم میکند. به این ترتیب، OTCL میتواند از اشیاء کامپایل شده در C++ استفاده کند و اشیاء OTCL با هر شیء متناظرشان در C++ مطابقت می یابند.

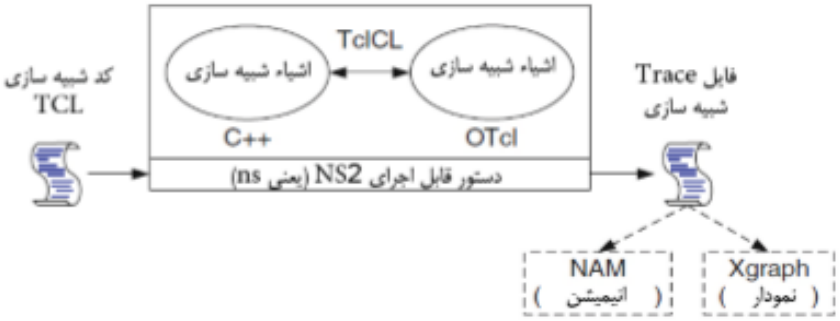
۱- Interpreter

۲- Object Oriented Tool Command Language: تعمیم یافته شی گرای زبان TCL که در سال ۱۹۹۷ توسط David Wetherall از دانشگاه MIT ایجاد شده است

۳- Network Animator

۴- Visualization

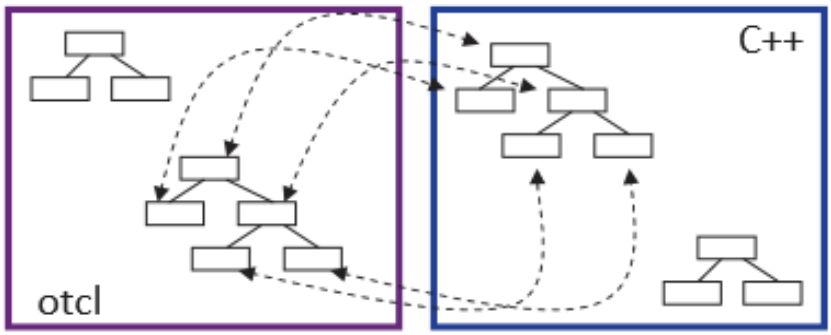
۵- TCL with classes: یک لایه چسبی از C++ روی OTCL ایجاد میکند



شکل ۱-۱: معماری کلی NS2

در اینجا دو کلاس زنجیره ای (سلسله مراتبی) وجود دارد: یکی زنجیره C++ کامپایل شده و دیگری زنجیره Otcl تفسیر (ترجمه) شده که یک تناظر یک به یک میان این دو برقرار است. (شکل ۲-۱)

طراحی شبیه ساز در واقع با جداسازی « داده[۱] » و « کنترل[۲] » صورت می گیرد. یعنی C++ برای داده (شامل پردازش در-بسته[۳]، هسته[۴] NS2، تسریع در اجراء حاوی جزئیات مفصل ، کنترل کامل) و OTCL برای کنترل (شامل پیکربندی[۵] سناریوی شبیه سازی ، عمل متناوب یا عمل فعال شده[۶] ، دستکاری اشیاء موجود C++، تسریع در تغییر و نوشتن).



شکل۲-۱: اشیای مجزای C++ و Tcl

زنجیره C++ کامپایل شده، امکان بازدهی بهتر و اجرای سریعتر شبیه سازی را فراهم میکند. این امر، به ویژه برای سنجش عملکرد پروتکلها و از طریق کاهش بسته ها[۷](ی داده) و کاهش زمان پردازش رخداد[۸]، موثر است.

در کد OTcl میتوان یک توپولوژی شبکه ای[۹] مشخص را پیاده سازی نمود که در آن رفتار انواع پروتکلها[۱۰]، کاربردها[۱۱]، گره[۱] ها و... قبلا در کلاسهای C++ تعریف شده اند.

۱- Data
۲- Control
۳- Per Packet Processing
۴- Core of ns
۵- configuration
۶- triggered or Periodic action
۷- Packet
۸- event processing time
۹- Network Topology: شکل و نحوه استقرار شبکه که شامل وسایل شبکه، محل قرار گرفتن آنها و تکنولوژی انتقال می باشد.
۱۰- Protocol
۱۱- Application

در NS2، به عنوان یک شبیه ساز رخداد–گسسته [۲]، پیشرفت زمان بستگی به تنظیم رخداد[۳]هایی دارد که به وسیله زمانبند [۴] کنترل میشوند. رخداد، یک شیء در C++ است که یک شناسه (ID) منحصر بفرد دارد و به وسیله زمانبند و یک اشاره گر شیء کنترل میشود.

کارکردهای NS2

بطور کلی کارکردهای **NS2** عبارتند از:

شبیه سازی شبکه های باسیم یا بی سیم یا ماهواره ای، مسیر یابی[۵] ، انتقال (از طریق TCP ، UDP، Multicast، Unicast)، منابع ترافیکی (Telnet, FTP, Web)، ضوابط صف بندی، کیفیت خدمات[۶] (Diffserv, Intserv)، شبکه های حسگر [۷]، ردیابی[۸] اطلاعات شبیه سازی شبکه، تولید آماری، تولید اعداد تصادفی و نمونه سازی برای سیستم های باسیم.

NS2 برای شبکه های بی سیم، امکان مسیریابی ویژه (غیر عمومی [۹]) و تعریف IP سیار [۱۰](یا گره سیار [۱۱]) را فراهم می کند.

مزایا و معایب NS2

مهمترین مزایای NS2 عبارتند از:

- قابلیت پیکر بندی آسان به دلیل استفاده از دو زبان برنامه نویسی مختلف (OTCL, C++)
- بسیاری از پروتکل ها قبلاً در آن پیاده سازی شده اند
- مدل های در دسترس بی شماری دارد
- مدل های سیار واقع گرایانه ای ایجاد میکند
- بهترین گزینه برای کسانی است که علاقمند به سطح دقت بالا در لایه های فیزیکی (PHY) هستند
- در شبیه سازی های بزرگ مقیاس (که معمولاً در این موارد از شبیه سازی مرحله ای استفاده می شود) کاربرد دارد
- قابلیت استفاده در شبیه سازیهای موازی [۱۲]
- بسیار شناخته شده است

۱- Node

۲- Discrete- Event Simulator

۳- event

۴- Scheduler

۵- Routing

۶- QOS: Quality Of Services

۷- sensor network

۸- trace

۹- Ad hoc routing

۱۰- Mobile IP

۱۱- Mobile Node

۱۲- در این نوع شبیه سازیها، دو (یا چند) شبیه ساز مختلف با هم ترکیب شوند و بطور همزمان دو نمونه از برنامه شبیه سازی سری (Serial) معمولی را روی پردازشگر های مختلف اجرا می کنند. این نمونه ها مستقل از یکدیگر اجرا می شوند و بصورت پیوسته، پارامترهای مدل شبیه سازی که مشاهده می کنند بر می گردانند یعنی در واقع چند پاسخ بصورت موازی (MRIP: Multiple Replication in Parallel) برگردانده میشود؛ و این همان چیزی است که پردازش کنترلی مرکزی (Central Controlling Process) نیاز دارد. در حال حاضر پروژه هایی مثل پروژه Akaroa2 (که شبیه سازی موازی را با NS2 یا OMNET++ انجام میدهد) وجود دارند که از این قابلیت شبیه سازها استفاده میکنند؛ با اینکار، افزایش سرعت و قابلیت اطمینان را برای شبیه سازیها به ارمغان می آورند.

- گروه های کاربران متعددی از آن استفاده میکنند
- Open source بوده و رایگان است

مهمترین معایب NS2 عبارتند از:

- مدت زمان طولانی برای آشنایی با آن
- سورس کد و دستور العملهای آن بد مستند سازی شده اند؛ مستندات آن برای افراد تازه کار مناسب نیست و تنها برای کسانی که با این شبیه ساز کار کرده اند کارآ است
- دشواری در ارزیابی سریع یک ایده کوچک (شما باید تمام ساختار های شبیه سازی را بدانید حتی اگر بخواهید قسمتی از پشته های پروتکل را شبیه سازی کنید.)

برای شروع کار با شبیه ساز NS2 لازم است تا آشنایی مختصری درباره زبان TCL حاصل شود. از این رو در ادامه به مقدمات زبان TCL پرداخته میشود.

زبان TCL

الف– آشنایی با TCL/TK

TCL [۱]، (با تلفظ تیکل Tickle)، به عنوان یک زبان چسبی (که مثل چسب به دیگر برنامه ها می چسبد!) ، کاربران را قادر می سازد تا برنامه ها و برنامه های کمکی[۲] را کنترل کنند. TCL و مکمل گرافیکی آن یعنی TK [۳] (با تلفظ تی کی tee kay) در سال ۱۹۸۰ توسط دکتر "جان آستِرهاوت"[۴] در دانشگاه برکلی کالیفرنیا ایجاد شدند. ایجاد برنامه TCL توسط دکتر آستِرهاوت، دو هدف عمده داشت:

۱- استفاده به عنوان زبان اسکریپتی[۵] ، که برنامه ها را قادر می سازد تا با احضار دستورات TCLبا یکدیگر ارتباط داشته باشند.
۲- استفاده به عنوان مفسر (مترجم) قابل جاسازی[۶] در برنامه های دیگر، که کاربران را قادر می سازد تا به کمک زبان اسکریپتی TCL، برنامه ها(یی که به زبانهای دیگر نوشته شده اند) را به دلخواه خود تغییر دهند. چرا که کار با برنامه TCL، به مراتب ساده تر از کار با برنامه های دیگر است.

TCL، تقریبا مشابه "ویژوال بیسیک برای برنامه های کاربردی[۷] " (VBA) است. همانطور که به کمک VBA میتوان از کارکرد برنامه های World، Excel و PowerPoint در یک برنامه کاربردی دیگر استفاده کرد، به وسیله TCL (و TK) نیز میتوان از مجموعه متنوعی از برنامه ها در یک برنامه دیگر بهره برد.

۱- Tool Command Language: زبان دستور- ابزار
۲- Utility
۳- (graphical) Toolkit
۴- Dr. John Ousterhout
۵- scripting language: یک زبان برنامه نویسی ساده که برای انجام کارهای خاص یا محدود طراحی میشود و اغلب با یک برنامه کاربردی خاص مرتبط است.
۶- embeddable interpreter
۷- Visual Basic for Applications (VBA): زبان ماکرونویسی که نگارشی از ویژوال بیسیک بوده و محصول شرکت مایکروسافت است.

همانطور که گفته شد، هدف اصلی TCL به عنوان یک زبان سطح بالا، فراهم کردن امکان تعامل میان برنامه های مختلف بود. اما از مدتها پیش، مثل هر برنامه یا زبان برنامه نویسی موفق دیگر، TCL/TK نیز از مرز هدف اصلی خود تجاوز کرده و گاها برنامه هایی به کمک آن نوشته میشوند که صدها و هزاران خط کد را در بر میگیرند ؛ قابلیتهای زبان هسته TCL تا جایی رشد کرده که کاربران این زبان قادرند برنامه های شبکه ای توانمندی را خلق کنند که میتوانند با پایگاه داده ها تعامل داشته باشند، وب را جستجو و در آن گشت و گذار کنند، به صفحات وب خدمت رسانی کنند و ابزار MIDI [۱] را کنترل کنند.

قابل ذکر است که TCL یک برنامه چند-محیطی [۲] است، به این معنی که کدهای TCL که مثلا در محیط لینوکس نوشته می شوند در هر محیط دیگری (مثلا ویندوز)، که مفسر (مترجم) TCL در آن وجود دارد، بدون هیچ گونه تغییری اجرا میشود. بنابراین جاوا اولین زبانی نیست که ادعا میکند "یک بار بنویس، همه جا اجرا کن! [۳] ". قابلیت چند محیطی بودن TCL، باعث می شود که، به عنوان مثال، کاربران مجبور به یادگیری نحوه تعامل با فایلهای لینوکس یا پشته پروتکلی TCP/IP نباشند.

TK در ساده ترین تعریف، امتداد زبان TCL (یا به عبارت دقیقتر، یک کتابخانه TCL) است و در واقع یک ابزار کمکی بمنظور خلق و استفاده از واسطهای گرافیکی کاربر [۴] می باشد. TK شامل دستوراتی برای ایجاد دکمه ها، جعبه های متنی [۵] و دیگر واسطهای گرافیکی) و همچنین کنترل رنگ و فونت است. با توجه به اینکه استفاده از این ابزار کمکی برای کار با شبیه ساز NS2 ضروری نیست، بیش از این به تشریح این ابزار پرداخته نمیشود. بنابراین در قسمت بعدی مستقیما به کار با برنامه TCL و دستورات مقدماتی آن تمرکز میشود.

با توجه به توضیحات گفته شده، ویژگیهای TCL از این قرارند:

- امکان توسعه سریع را فراهم میکند
- از واسط گرافیکی بهره میبرد
- با بسیاری از محیطها سازگار است
- انعطاف پذیر است؛ بنحوی که میتوان آن را به سادگی در برنامه های دیگر استفاده کرد
- استفاده از آن ساده است
- رایگان است

ب- دستورات مقدماتی TCL

۱- **درج توضیحات**: برای درج توضیحات [۶] در زبان TCL از کاراکتر # (number sign یا pound sign یا sharp) استفاده می شود. بنابراین عباراتی که در جلوی این کاراکتر نوشته میشوند توسط مترجم TCL تفسیر نمیشوند.

```
#so we are going to start programming with TCL
```

۲- **اعلان متغیر**: جهت تعریف متغیرها در این زبان از کلمه کلیدی set استفاده می شود. به عنوان مثال اگر بخواهیم مقدار اولیه ۱۲ را در متغیر a ذخیره کنیم از دستور

```
set a 12
```

استفاده میکنیم. بنابر این این عبارت معادل عبارت a=12 (در زبان C) است.

۱- Musical Instrument Digital Interface: رابط دیجیتالی ادوات موسیقی، فایل های MIDI جهت برقراری ویدئو کنفرانس ها و پخش فیلم در اینترنت به کار می روند

۲- cross-platform: اصطلاحی برای یک نرم افزار کاربردی (یا سخت افزار) که در بیش از یک محیط قابل اجرا (استفاده) است.

۳- Write once, run anywhere

۴- graphical user interface

۵- text box

۶- comment

تذکر: TCL یک زبان رشته–محور [1] است؛ به این معنی که تمام متغیرها در این زبان از نوع رشته (string) هستند. حتی اعداد! در قسمتهای بعدی متوجه خواهیم شد که چگونه مترجم TCL عملیات ریاضی روی اعداد را (به عنوان عبارت ریاضی و نه به عنوان رشته) تشخیص خواهد داد.

مثال: اگر بخواهیم رشته hello را در متغیر b ذخیره کنیم مینویسیم:

```
set b hello
```

در صورتی که رشته مفروض *بیش از یک کلمه* باشد باید از علامت کوتیشن مضاعف [2] یعنی " در ابتدا و انتهای رشته استفاده کنیم؛

مثلا اگر بخواهیم رشته hello world را در متغیر c ذخیره کنیم مینویسیم:

```
set c "hello world"
```

تذکر ۱: در مورد رشته های تک کلمه ای هم میتوان از کوتیشن مضاعف استفاده کرد:

```
set b "hello"
```

تذکر ۲: اگر یک رشته طولانی داشته باشیم که میان کلمات آن فاصله (space) وجود نداشته باشد، الزامی در استفاده از کوتیشن مضاعف نیست:

```
set c helloWorld
```

تذکر ۳: برای حذف یک متغیر از دستور unset استفاده می شود:

```
unset c
```

در این حالت، و پس از اجرای این فرمان، متغیر c حذف می شود و دیگر متغیری به نام c وجود نخواهد داشت.

۳- **نمایش مقدار متغیرها**: در مثالهای قبل دیدیم که چگونه مقدار ۱۲ در متغیر a ذخیره میشود؛ حال میخواهیم ببینیم چگونه میتوان به مقدار ذخیره شده در متغیر a دسترسی یافت. برای دسترسی به مقدار یک متغیر از علامت دلار یعنی $ استفاده میشود. مثلا اگر بخواهیم **مقدار** متغیر a را در متغیر d ذخیره کنیم، مینویسیم:

```
set d $a
```

یعنی **مقدار** a را در d قرار بده! این عبارت معادل عبارت d=a (در زبان C) است. در این حالت مقدار متغیر d برابر ۱۲ میشود.

به عبارت کلی هر جا بخواهیم از **مقدار** یک متغیر استفاده کنیم (چه در عبارات ریاضی، چه در مقدار دهی متغیرهای دیگر، چه در هنگام چاپ خروجی و ...)، از علامت $ استفاده میکنیم.

۴- **چاپ نتایج در خروجی**: برای چاپ نتایج در خروجی از کلمه کلیدی puts استفاده میشود. مثلا اگر بخواهیم مقدار متغیر a را در خروجی چاپ کنیم، می نویسیم:

```
puts $a
```

در این حالت مقدار ۱۲ در خروجی چاپ می شود. و اگر داشته باشیم:

```
puts "a is $a"
```

در این حالت عبارت a is 12 در خروجی چاپ میشود.

تذکر: به جای علامت " میتوان از کروشه برای مجموعه داده ها استفاده کرد:

```
set c {hello world}
```

که معادل عبارت set c "hello world" است.

تفاوت استفاده از کروشه و استفاده از کوتیشن مضاعف با مثال زیر مشخص می شود:

۱- string-based

۲- double quotation

حالت ۱ : puts “c is $c”

خروجی : c is hello world

حالت ۲ : puts {c is $c}

خروجی : c is $c

همانطور که مشاهده می شود، در حالت اول که از علامت " برای مجموعه ای از رشته ها استفاده کرده ایم، عبارت $c به عنوان **مقدار متغیر** c قلمداد می شود در حالی که در حالت دوم که از علامت } برای مجموعه ای از رشته ها استفاده کرده ایم، عبارت $c صرفا به عنوان یک رشته تفسیر شده است.

۵- **عبارات ریاضی**: همانطور که پیشتر گفته شد، تمام متغیرها در زبان TCL از نوع رشته هستند. اما چگونه میتواند روی اعداد (که آنها هم به عنوان رشته در نظر گرفته میشوند) عملیات ریاضی انجام داد؟ این کار با دستور expr که از واژه "expression" به معنی "عبارت" (و در اینجا بمعنی عبارت ریاضی) گرفته شده، انجام می شود. به عنوان مثال برای چاپ خروجی جمع دو عدد ۵ و ۱۰ به این ترتیب عمل میکنیم:

```
puts [expr 5+10]
```

خروجی: ۱۵

دقت کنید که عبارت ریاضی به همراه کلمه کلیدی expr حتما باید درون علامت براکت [] نوشته شوند.

مثال: برنامه ای که عدد ۳ را در متغیر a و عدد ۷ را در متغیر b قرار داده، این دو متغیر را در هم ضرب کرده و حاصل را در متغیر c قرار داده و نهایتا مقدار c را در خروجی چاپ میکند:

```
#define a=3 and b=7
   set a 3
   set b 7
#now define c=a*b
   set c [expr {$a*$b}]
    puts “c is $c”
```

خروجی: c is 21

در کد فوق میتوان به جای عبارت expr{$a*$b} از expr($a*$b) یا expr $a*$b استفاده کرد اما بهتر است بمنظور یکدستی برنامه (چنانچه در ادامه نیز خواهید دید)، مجموعه داده ها با { و } مشخص شوند.

تذکر: در زبان TCL نوع اعداد (اعشاری یا صحیح) با توجه به مقادیر متغیرها تعریف میشوند. به مثال زیر توجه کنید:

```
set a [expr 1/5]
```

در این حالت مقدار متغیر a برابر 0 (صفر) میشود. اگر بجای عبارت فوق بنویسیم:

```
set a [expr 1.0/5.0]
```

در این حالت مقدار متغیر a برابر 0.2 میشود.

۶- **دریافت مقادیر از ورودی**: برای دریافت یک مقدار از ورودی از دستور gets stdin استفاده میشود. به عنوان مثال پس از اجرای کد زیر:

```
gets stdin c
```

TCL منتظر میماند تا مقداری وارد شود و کلید enter فشرده شود. در این حالت مقدار وارد شده در متغیر c ذخیره میشود.

مثال: برنامه ای که با چاپ پیغام مناسب، یک اسم را از کاربر دریافت کرده و اسم وارد شده را به همراه تعداد کاراکترهای آن چاپ میکند:

```
puts -nonewline "Please enter name: "
flush stdout
set count [gets stdin playerName]
puts "Player name is $playerName."
puts "It has $count characters."
```

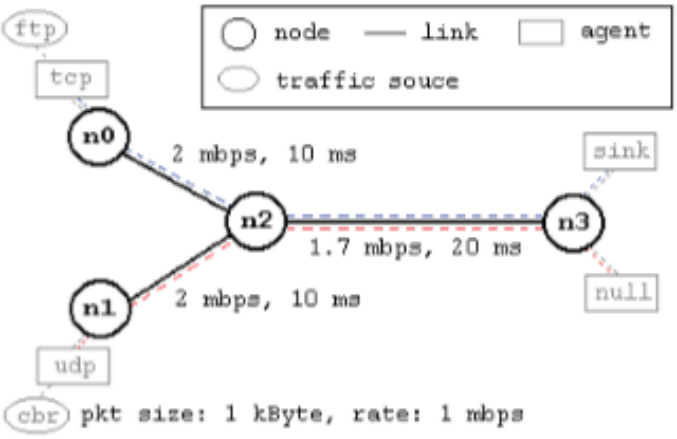
تذکر ۱: عبارت nonewline– جهت گرفتن نام کاربر در همان سطر نمایش پیغام "Please enter name: " استفاده شده است.

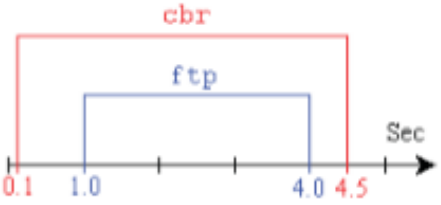
تذکر ۲: دستور flush stdout برای خالی کردن قسمتی از حافظه (بافر [۱]) استفاده شده است. هنگامی که در یک سطر هم پیغام چاپ میکنیم و هم ورودی میگیریم باید از این دستور استفاده کنیم؛ تا فضای حافظه مورد نیاز در اختیارمان قرار بگیرد.

تذکر ۳: پس از وارد کردن هر رشته بصورت ورودی، تعداد کاراکترهای آن رشته نیز بصورت خودکار وارد چاپ میشود؛ بنابراین نیازی به نوشتن دستور مجزا برای محاسبه تعداد کاراکترهای یک رشته نمیباشد.

۲ -۳ – مثالی برای آشنایی با نحوه کار با NS2 :

در این مثال می خواهیم یک شبکه با ۴ نود به نام های n1 ، n2 ، n3 و n4 پیاده سازی کنیم که لینک ارتباطی بین n0 و n2 و همچنین n1 و n2 از نوع duplex با پهنای باند ۲ مگابیت در ثانیه و تاخیر ۱۰ میلی ثانیه است و لینک duplex بین n2 و n3 دارای پهنای باند ۱.۷ مگابیت در ثانیه و تاخیر ۲۰ میلی ثانیه است .





هر نود از یک صف از نوع DropTail با حداکثر اندازه ۱۰ استفاده میکند.

N0 حاوی یک عامل tcp و n1 حاوی یک عامل tcp-sink است .

عامل های شبکه نشانگر نقاط انتهایی یک اتصال لایه شبکه اند که بسته های این لایه تولید کرده و در طرف دیگر تحویل لایه متناظر می دهند . بخشی از عامل توسط OTCL و بخشی از آن به زبان C++ طراحی شده است . عامل در پیاده سازی پروتکل های لایه‌ای مختلف استفاده می‌شود . (به طور پیش فرض اندازه پکت هایی که یک عامل tcp میتواند تولید کند 1 کیلو بایت است .یک عامل tcp sink پکت های Ack تولید و به فرستنده پیام ارسال میکند و پکت هاب دریافت شده را آزاد میکند .)

جزئیات بیشتر را در حین نوشتن اسکریپت ارائـه خـواهم داد .خـوب نوشـتن اولـین اسـکریپت Tcl بـرای پیـاده سـازی توپولوژی شکل ۱-۱ را شروع میکنیم:

میتوانید اسکریپت هایی Tcl را در هر برنامه ویرایشگر متن بنویسید.

Nano ns_example.tcl

```
#Create a simulator object
set ns [new Simulator]
#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red
#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf
#Define a 'finish' procedure
proc finish {} {
        global ns nf
        $ns flush-trace
        #Close the NAM trace file
        close $nf
        #Execute NAM on the trace file
        exec nam out.nam &
        exit 0
}
#Create four nodes
set n0 [$ns node]
```

```
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail
#Set Queue Size of link (n2-n3) to 10
$ns queue-limit $n2 $n3 10
#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
#Monitor the queue for link (n2-n3). (for NAM)
$ns duplex-link-op $n2 $n3 queuePos 0.5
#Setup a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
#Setup a FTP over TCP connection
```

```
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2
#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false
#Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr stop"
#Detach tcp and sink agents (not really necessary)
$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"
```

```
#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"
#Print CBR packet size and interval
puts "CBR packet size = [$cbr set packet_size_]"
puts "CBR interval = [$cbr set interval_]"
#Run the simulation
$ns run
```

و برای اجرای آن دستور زیر را اجرا کنید :

```
ns  ns_example.tcl
```

توضیحات ns_example.tcl :

قبل از هر کار در هر اسکریپت tcl برای NS2 باید یک شیئ شبیه ساز ایجاد کرد. این کار با دستور زیر انجام می شود

```
set ns [new Simulator]
```

این دستور یک شیئ شبیه ساز NS را ایجاد می کند و آنرا به متغیر NS منصوب می کند .

با استفاده از روال ها و خصوصیات این شیئ می توان توپولوژی شبکه را پیاده سازی کرد.

به عنوان مثال شیئ شبیه ساز دارای متدهایی است که عملیات زیر را انجام می دهند :

- گره ها و خطوط ارتباطی بین آنها را ایجاد می کند .
- اشیا مربوط به عناصر شبکه (با 13dged13é-agent) را به یکدیگر متصل می کند .
- ارتباط بین منابع ترافیکی و دریافت کننده ها را برقرار می کند .
- پارامتر های نمایش توسط نرم افزار NAM را مشخص می کند .

(NAM ابزاری برای نمایش گرافیکی شبیه ساز NS است و محیطی را در اختیار می گذارد که بتوان با استفاده از آن حرکت واقعی بسته های داده ای را مشاهده کرد.)

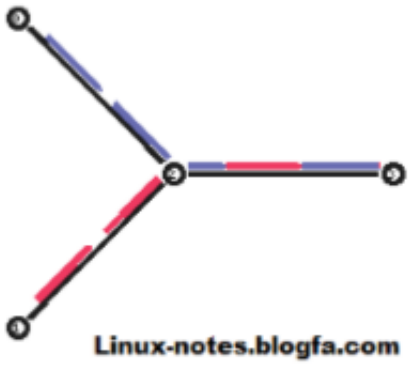
$ns color 1 Blue

$ns color 2 Red

این خطوط برای تعیین رنگ جریان پکت ها استفاده می شود. Syntax کلی آن به صورت زیر است :

$ns color fid color

در اینجا جریان پکت ها با fid شماره ۱ با رنگ آبی و جریان پکت ها با fid شماره ۲ با رنگ قرمز نمایش داده می شود.



این بخش برای تنظیم محیط گرافیکی نمایش شبیه ساز (NAM) است و تاثیری در شبیه سازی ندارد .

NAM برای داده های ترسیم نیاز به یک فایل باز دارد که در اولین خط کد این بخش فایل out.nam رابرای نوشتن باز کرده و نام nf را به آن می دهیم . در خط دوم مشخص میکنیم که شیئ شبیه ساز ، همه ی داده های شبیه سازی که به NAM مربوط می شود را در فایل out.nam بنویسد.

Set nf [open out.nam w]

$ns namtrace-all $nf

تعریف روال پایان :

```
proc finish {} {
    global ns nf
    $ns flush-trace
    #Close the NAM trace file
    close $nf
    #Execute NAM on the trace file
    exec nam out.nam &
    exit 0
}
```

یک روال پایان تعریف کردیم تا در خاتمه کار با فراخوانی آن از شبیه ساز خارج شویم.

در این مرحله ۴ نود شبکه ایجاد می کنیم . هر نود را با ایجاد یک شیئ جدید با دستور $node تولید کرده و آنرا با دستور set به متغیر های n0 تا n4 اختصاص می دهیم.

```
Set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
```

اکنون نوبت به ایجاد لینک ارتباطی بین نود ها رسیده است :

```
$ns duplex-link $node1 $node2 Bandwidth Delay Queue-type
```

پس لینک های ارتباطی را با مشخصات داده شده (پهنای باند – تاخیر – نوع صف) تعریف می نماییم.

```
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.7Mb 20ms DropTail
```

همچنین می توان اندازه اندازه صف هر لینک را محدود کرد :

```
$ns queue-limit $n2 $n3 10
```

اینجا اندازه لینک ارتباطی بین نود های n2 و n3 را ۱۰ محدود کردیم .

برای بهبود شمای توپولوژی شبکه بهتر است مکان نودها را مشخص کرد .

```
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n2 $n3 orient right
```

برای مشاهده ی تاثیر این خطوط بر ظاهر شبیه ساز پیشنهاد میکنم اسکریپت را بدون این خطوط نیز اجرا کنید .

امکان مانیتور کردن پرقدرت در NS یکی از مزایای آن است . در NS میتوان صف هر لینک را نیز مانیتور کرد:

```
$ns duplex-link-op $n2 $n3 queuePos 0.5
```

حال می توانید بسته ها را در صف ببینید و مشاهده کنید کدام یک دور انداخته می شوند . در اینجا با صف از نوع Droptail فقط پکت های آبی دور ریخته میشود . برای مشاهده تاثیر نوع صف آنرا به SFQ تغییر دهید و دوباره آنرا اجرا کنید :

```
$ns duplex-link $n0 $n2 2Mb 10ms SFQ
$ns duplex-link $n1 $n2 2Mb 10ms SFQ
$ns duplex-link $n2 $n3 1.7Mb 20ms SFQ
```

تا این مرحله فقط سه نود و لینک ارتباطی بین آنها را ایجاد کرده و مکان آنها را در NAM مشخص کردیم ولی هنوز هیچ نوع داده ای برای تبادل بین نودها مشخص نشده است.

در NS همیشه داده از یک عامل (agent) به دیگری ارسال می شود . بنابراین گام بعدی ایجاد عامل های ارسال کننده (منابع ترافیکی) و دریافت کننده داده است.

برقراری یک اتصال TCP :

```
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
```

در این خطوط یک عامل TCP ایجاد کردیم. کاربران NS2 می توانند هر نوع عامل یا منبع ترافیکی را ایجاد کنند . در واقع عامل ها و منابع ترافیکی شیئ های اصلی در NS هستند که عمدتا در C++ پیاده سازی شده و در OTCL لینک می شوند. برای ایجاد عامل ها و منابع ترافیکی ، کاربر باید نام کلاس های این اشیا را بداند (Agent/TCP ، Agent/TCPSink ، Application/Ftp). این اطلاعات در مستندات NS موجود است ولی یک راه میانبر برای آن فایل

/ns-2/tcl/libs/ns-default.tcl

است. این فایل حاوی مقادیر پیش فرض پارامتر های قابل پیکربندی برای اشیا شبکه در NS است .بنابراین راهنمای خوبی برای مشخص کردن اینکه چه نوع شی های شبکه در NS قابل دسترس است و پارامتر های قابل تغییر در NS است .

می توانید پارامتر های مناسب برای هر نوع عامل را در صفحه راهنمای NS ببینید :

(http://www.isi.edu/nsnam/ns/doc/index.html)

روال attach-agent یک عامل ایجاد شده را به یک نود پیوند می دهد .

```
$ns attach-agent node agent
```

به عنوان مثال :

```
$ns attach-agent $n0 $tcp
```

قدم بعدی ساختن یک اتصال شبکه منطقی بین اتصالات است. کد زیر یک ارتباط شبکه را با تنظیم آدرس مقصد و گروه آدرس پورت برای هر شبکه برقرار می کند .

```
$ns connect agent1 agent2
```

به عنوان مثال درکد زیر :

```
$ns connect $tcp $sink
```

ارتباط منطقی بین عامل های tcp و sink برقرار شده است .

در این مرحله منبع ترافیکی FTP را ایجاد می کنیم.

```
#Setup a FTP over TCP connection
```

```
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
```

ایجاد عامل udp و برقراری ارتباط از نوع udp بین n1 و n3

```
#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2
```

```
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false
```

این خطوط یک تولید کننده ترافیک CBR را به عامل UDP متصل میکند. (CBR مخففی است برای Constant Bit Rate)

اندازه بسته روی ۱۰۰۰ بایت تنظیم شده است و سرعت تولید ۱ مگابایت در ثانیه است.

خوب هم اکنون تنظیمات شبکه انجام شده است . قدم بعدی نوشتن یک سناریوی شبیه سازی است .

شیئ شبیه ساز توابع برنامه ریزی مختلفی دارد که مهمترین آن تابع time است :

$ns at time "String"

این شیئ برای زمانبندی شبیه ساز استفاده می شود .

$ns at 0.1 "$cbr start"

$ns at 1.0 "$ftp start"

$ns at 4.0 "$ftp stop"

$ns at 4.5 "$cbr stop"

CBR کار خود را در ثانیه 0.1 شروع می کند و در ثانیه 4.5 خاتمه می دهد و همچنین منبع ترافیکی ftp کار خود را در ثانیه 1.0 آغاز و در ثانیه 4.0 به اتمام می رسد .

در آخر کار عامل ها را از یکدیگر جدا می کنیم . اگر چه این کار لازم نیست .

$ns at 4.5 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

خط آخر نیز برای اجرای ns است .

نحوه نصب و راه اندازی NS2

ابتدا فولدر MPR Tutorial **روی هارد کپی میکنیم .**

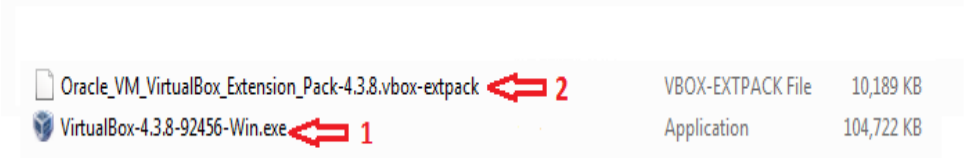
نرم افزار xmind **داریم داخل این مجموعه که نرم افزار مدیریتی و کاربردی است که می توانیم مثلا برای نصب یک نرم افزار از این برنامه استفاده کرد تمام مراحل را به صورت فلوچارتی برای مانمایش خواهد داد تمام نکات و سلسله مراتب بهش میدهیم و برای ما به صورت فلوچارت نمایش خواهدداد و طراحی کارهای مدیریتی را میتوانیم با این نرم افزار انجام دهیم مثلا برای مهندسی نرم افزار میتوانیم از این نرم افزار استفاده کنیم .**

فولدر بعدی به نام VirtualMachin **، توجه داریم که** NS2 **تحت لینکوس هست پس اینجا یک** image **داریم از این مجموعه تهیه شده است .**

یک فایل Readme **هم داریم که با رایت کلیک کردن و باز کردن توسط** Notpad **میتوانیم اطلاعات داخل آنرا مرتب شده مشاهده کنیم که** Username , password **داریم .**

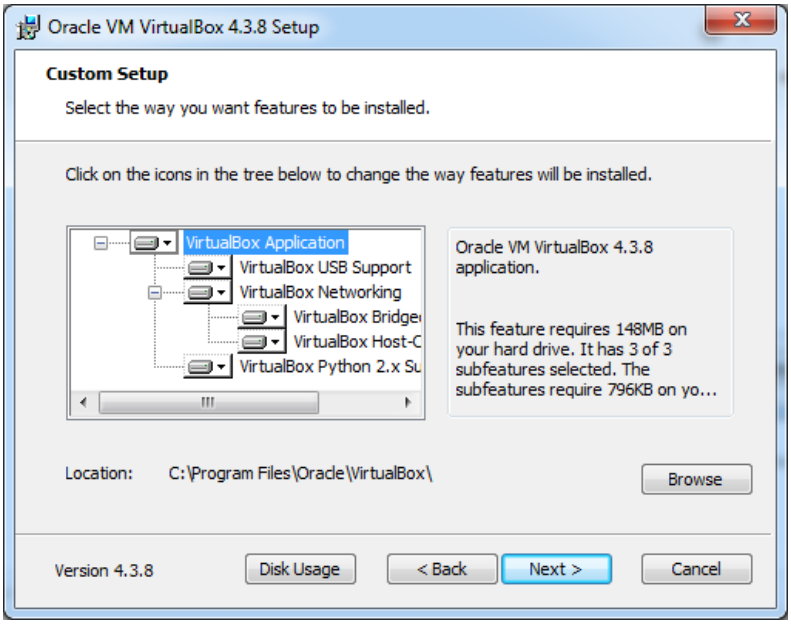
فولدر بعدی VirtualBox **است نسخه های لینکوس و ویندوز داریم .نسخه ویندوزی انتخاب میکنیم**

1 - ابتدافایلشماره 1 را اجراکنید.

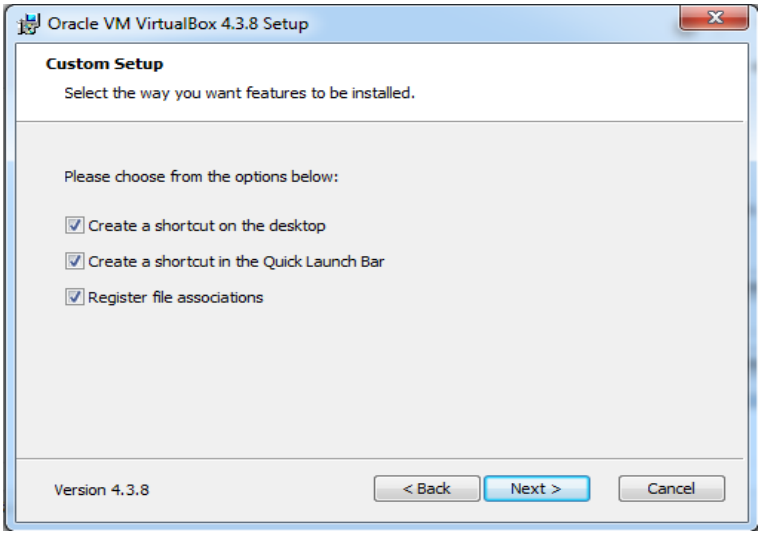


2 -بر روی Next **کلیک کنید.**

3 -بر روی زیر شاخه مورد نظر کلیک کنید تا ویژگی مورد نظر خود را انتخاب نمایید ، همچنین محلی که می خواهید برنامه شما نصب شود را تعیین کنید.

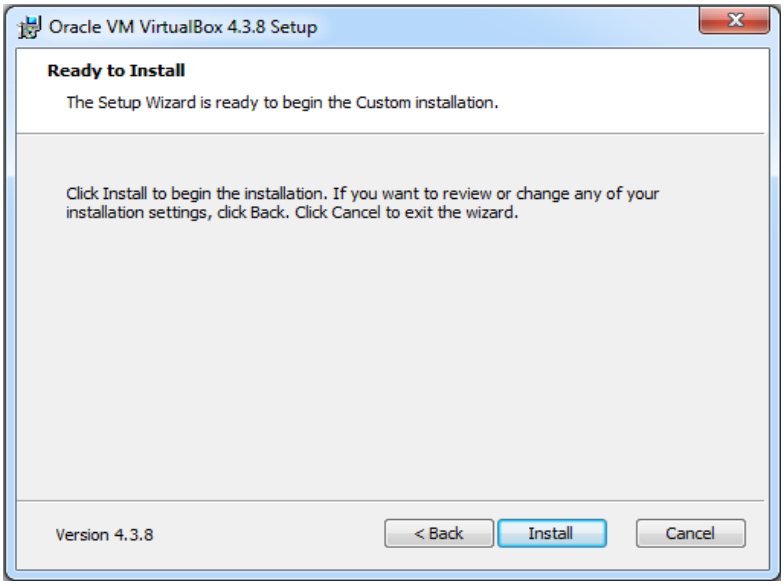


4 -محلی که می خواهید فایل شما ایجاد شود را انتخاب کنید.

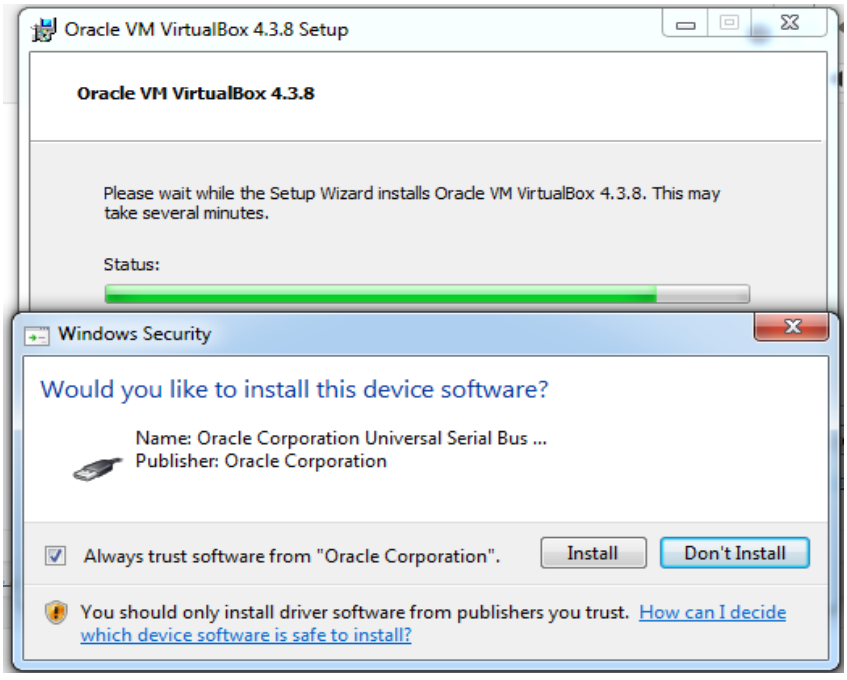


5 -در این مرحله از نصب پیغام میدهد که با نصب Network Connection ، VirtualBox**شما ریست میشود و موقتاً** Discconnect**میشوید.**

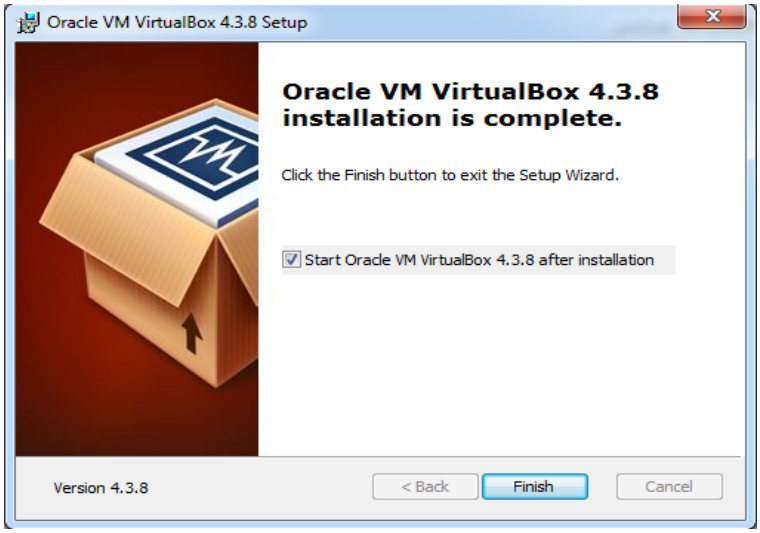
6 -Install را بزنید تا برنامه شروع به نصب کند.



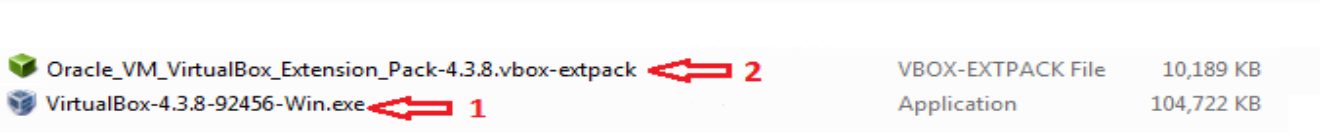
7 -در زمان نصب یک پنجره باز می شود در آن check Box را تیک بزنید و Install را کلیک نمایید تا ادامه نصب را انجام دهد.



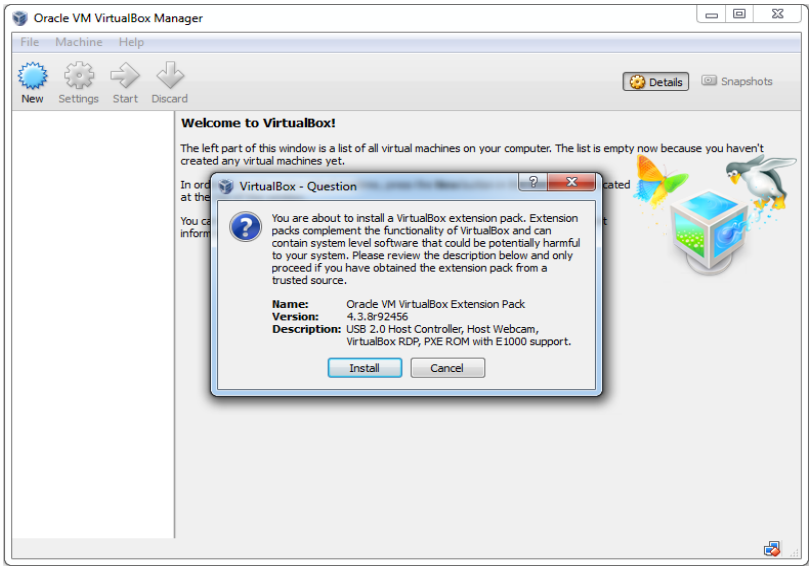
8 -با زدن Finish **نصب شما به پایان میرسد.**



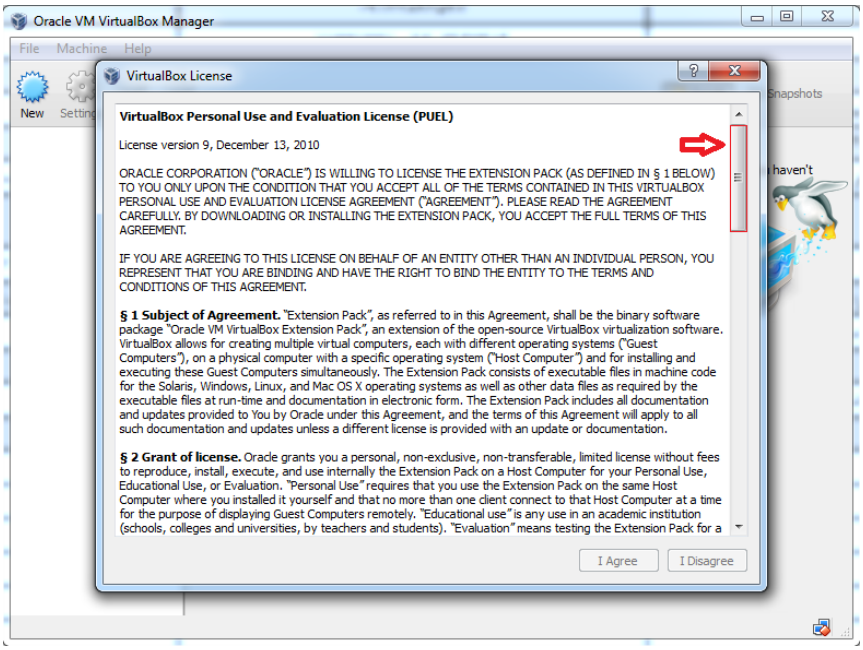
9 -اکنون فایل شماره 2 که در تصویر اول مشاهده نمودید به این شکل در آمده است :



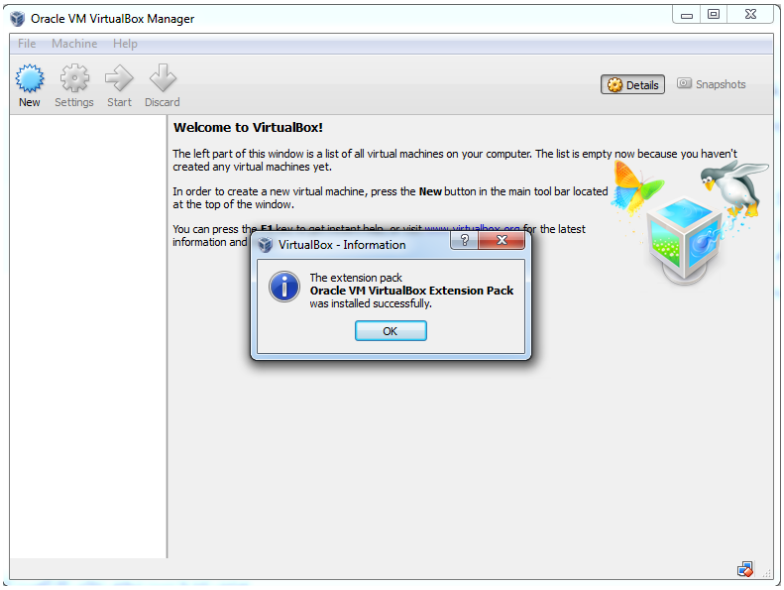
10 -بر روی فایل شماره 2 کلیک کنید و Install **را بزنید.**



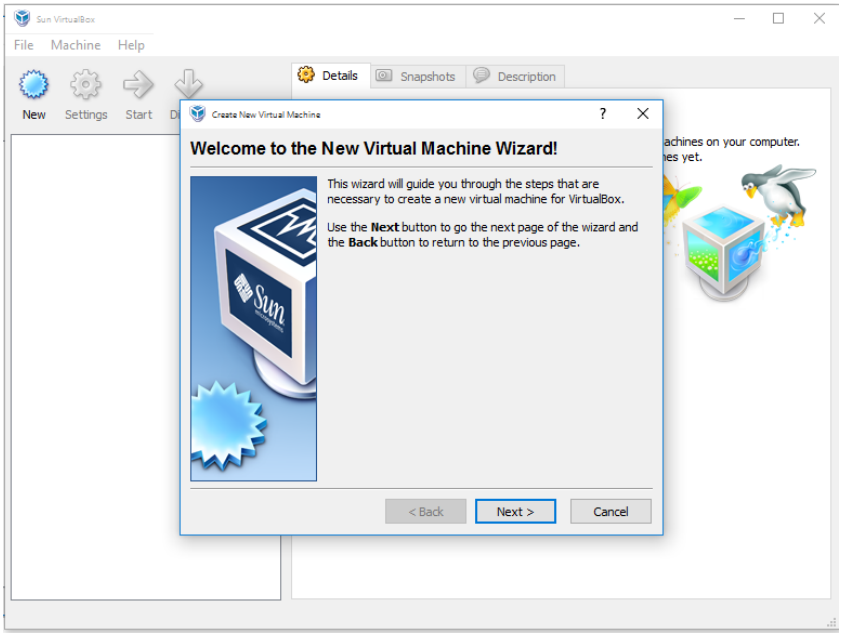
11 ScrollBar **-سمت راست را به سمت پایین تا انتهای توضیحات بکشید تا دکمه** I Agree **فعال شود.**



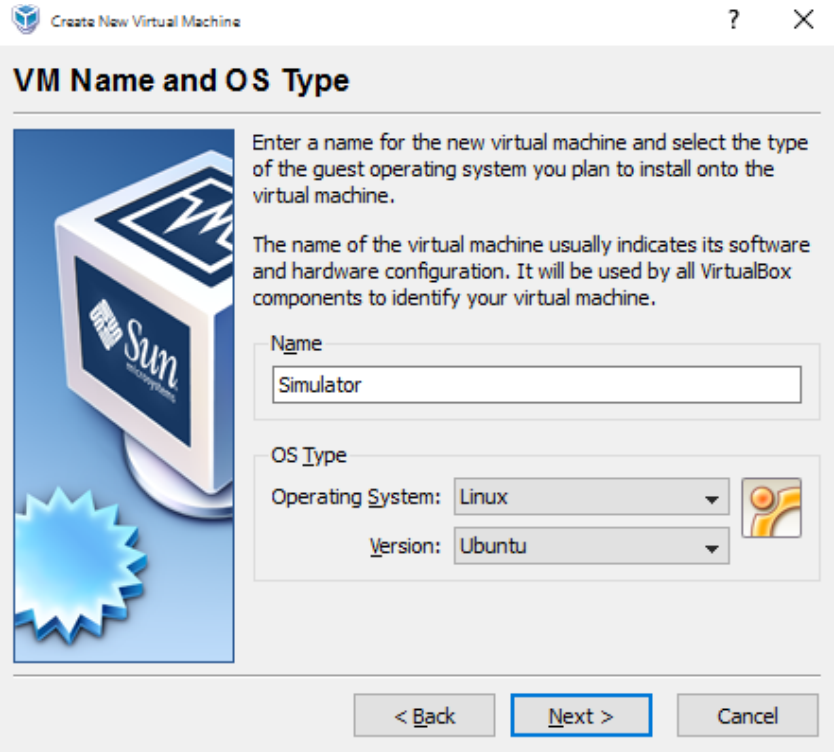
12 -با زدن I Agree **، پیغام میدهد با موفقیت برنامه شما نصب شده است.**



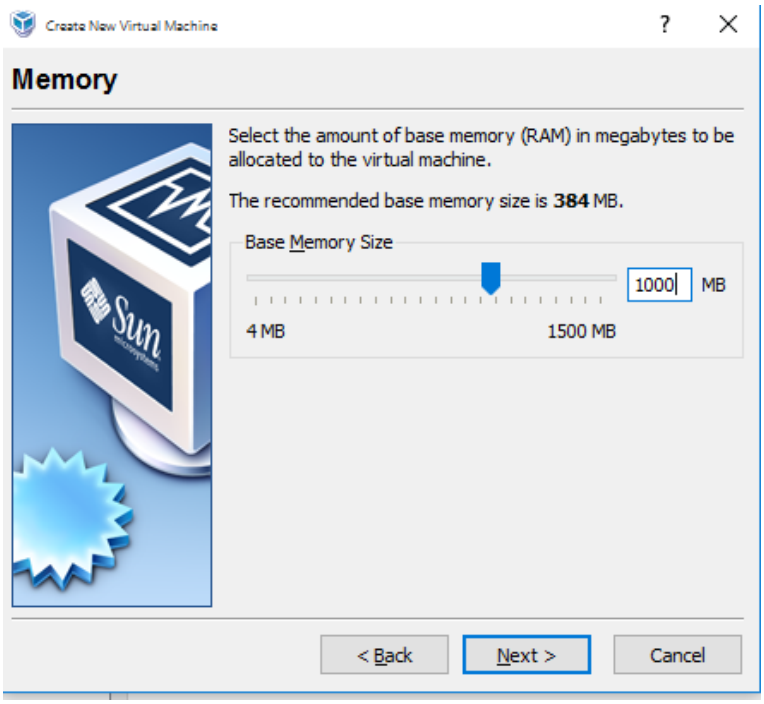
دراین مرحله با انتخاب دکمه New **مراحل ادامه میدهیم سپس** Next **انتخاب میکنیم :**



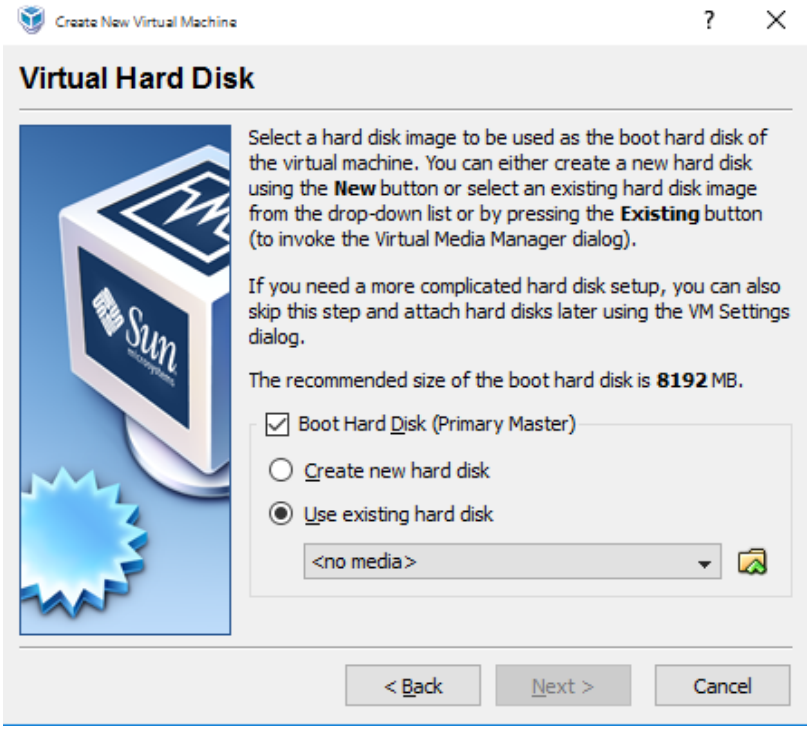
- **دراین مرحله** OS **و** Version **مشخص میکنیم**



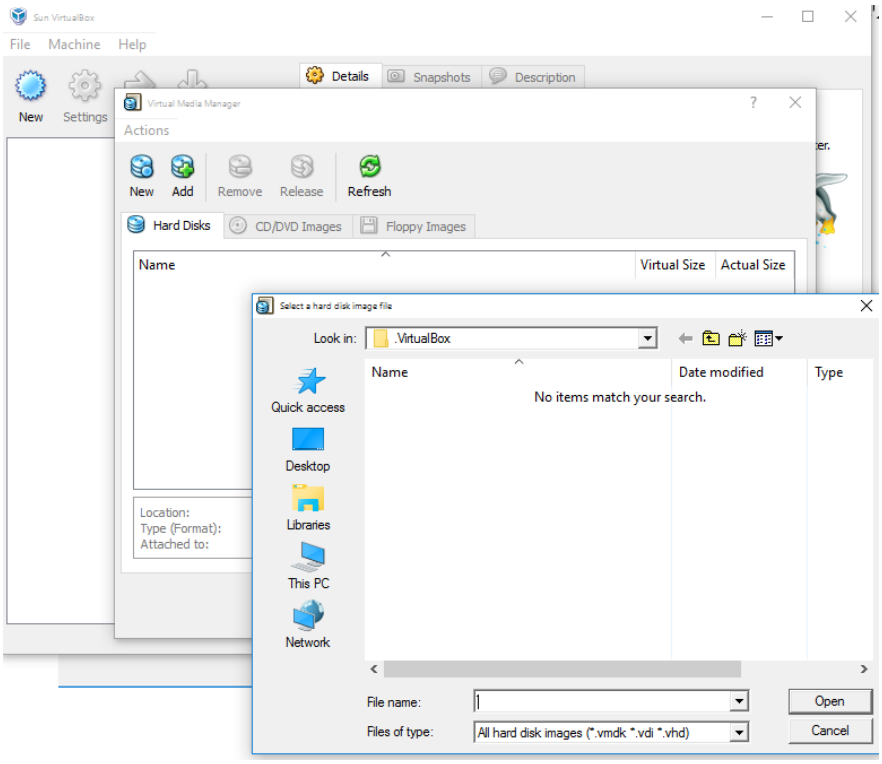
-Memory **را مشخص میکنیم**



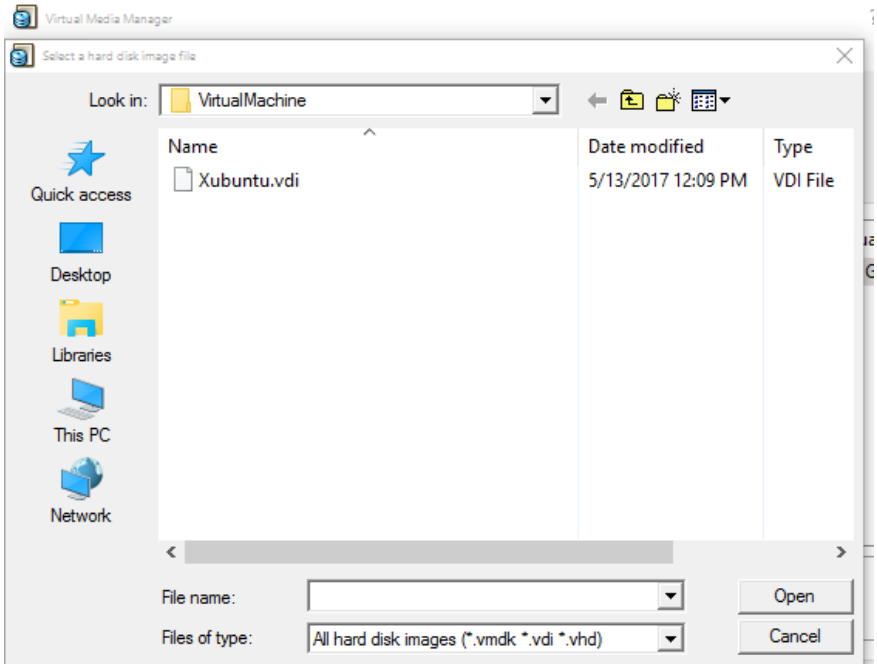
- **دراین مرحله گزینه دوم انتخاب میکنیم و گزینه** Browser **انتخاب میکنیم** :



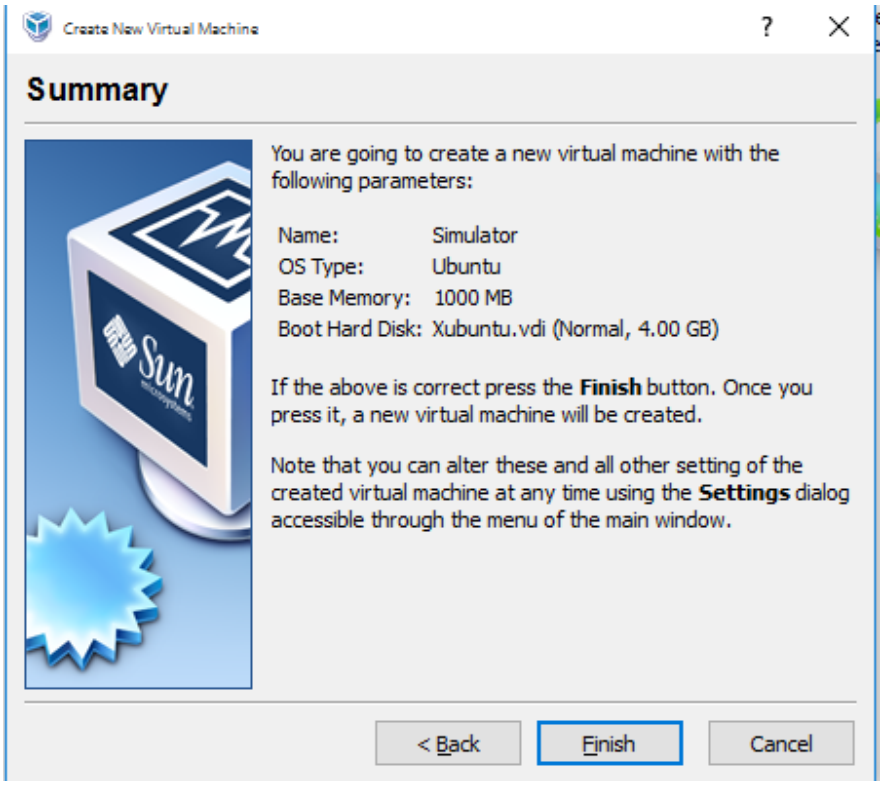
- **با انتخاب گزینه** browser **، کادر زیر نمایش داده میشود که گزینه** Add **انتخاب میکنیم** :



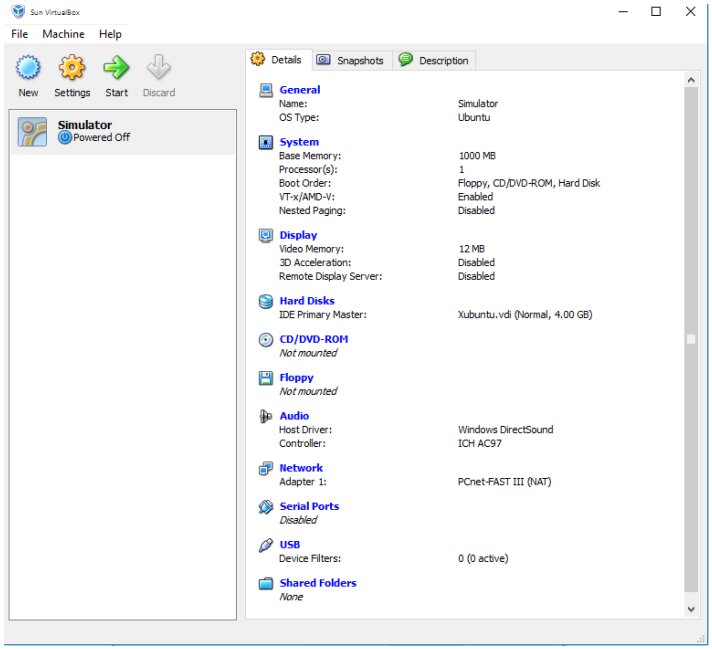
- **دراینجا فایل** Xubuntu.vdi **از مسیر انتخاب میکنیم:**



بعد از انتخاب فایل مورد نظر و بارگذاری با انتخاب دکمه Nex **وارد مرحله پایانی و نصب انجام خواهد شد:**



مراحل نصب که تمام شد گزینه Srart **انتخاب میکنیم :**



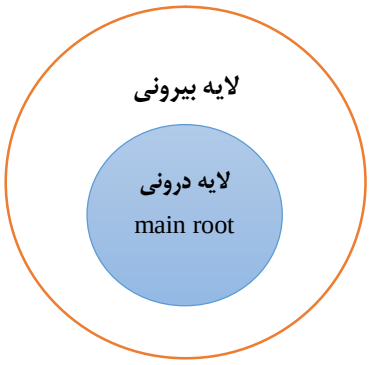
دراین مرحله محیط عملیاتی ما لینوکس خواهد شد :

username: hassan

password: EF-MPR

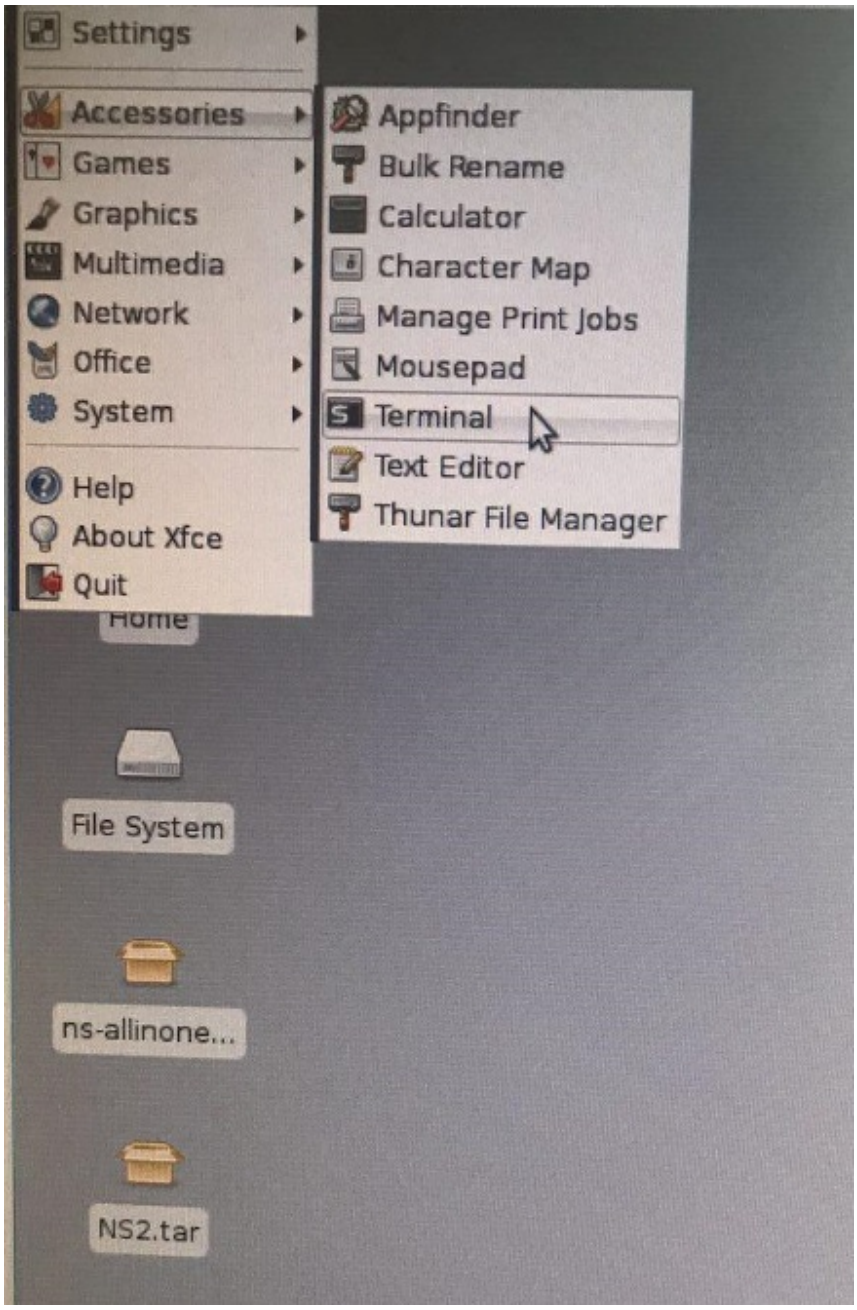
وارد بحث سیستم عامل لینوکس میشویم :

امنیت بالاتری نسبت به ویندوز دارد یک سیستم عامل پایدار است از لحاظ بحث کسب و کار ویندوز بهتر است چون کاربرپسند تر است اما لینوکس محیط گرافیکی مناسبی ندارد لینوکس2 لایه است لایه درونی امنیت بالاتری دارد هرچی روی لایه بیرونی نصب شود راحت اجرا میشود اما اگر در لایه درونی نصب شود باید دسترسی داشته باشیم و باید پسورد وارد کنیم برای دسترسی باید دستور su **یا** sudo **وارد کنیم برای دسترسی به لایه درونی و خودمان دستور نصب میدهیم کجانصب شود**



از مسیر Application / accessories/terminal **وارد شده و اگر اینجا دستور** NS **بزنیم پیغام میدهد که وجود ندارد**

بنابراین SU **یا** SUDO **وارد میکنیم** :



بعد از وارد کردن دستور su **از ما پسورد میخواهد که وارد میکنیم بعدعلامت % نمایش داده میشود وارد میکنیم**