

۲	فهرست شکل‌ها.....
۳	۱. محیط شبیه سازی.....
۵	۱-۱. تعریف.....
۵	۲-۱. ساختار.....
۶	۱-۲-۱. هسته OverMelon.....
۷	۱-۲-۱. Engine Manager.....
۷	۲-۱-۲-۱. Engine Interface.....
۸	۳-۱-۲-۱. Entity.....
۱۱	۴-۱-۲-۱. Log.....
۱۲	۵-۱-۲-۱. Preferences.....
۱۳	۶-۱-۲-۱. Network.....
۱۹	۲-۲-۱. ناظر.....
۲۰	۲-۲-۲-۱. Supervisor Manager.....
۲۰	۳-۲-۲-۱. Supervisor Handler.....
۲۱	۴-۲-۲-۱. Application.....
۲۱	۵-۲-۲-۱. GUI.....
۳۰	پیوستها.....
۳۳	منابع و مراجع.....

فهرست شکل‌ها

- شکل ۱-۱ شبیه ساز OverMelon؛ نمای اجزای کلی ۵
- شکل ۲-۱ مقایسه لایه های OSI با OverMelon ۶
- شکل ۳-۱ شبیه ساز OverMelon؛ ساختار جزئی ۷
- شکل ۴-۱ شبیه ساز OverMelon؛ نمای ساختار Engine Interface ۸
- شکل ۵-۱ شبیه ساز OverMelon؛ ساختار جزئی Entity ۸
- شکل ۶-۱ شبیه ساز OverMelon؛ نمای کلی ارتباط میان ماژول های Entity ۱۰
- شکل ۷-۱ شبیه ساز OverMelon؛ ساختار جزئی Log ۱۲
- شکل ۸-۱ شبیه ساز OverMelon؛ ساختار جزئی Preferences ۱۳
- شکل ۹-۱ شبیه ساز OverMelon؛ ساختار جزئی Network ۱۴
- شکل ۱۰-۱ شبیه ساز OverMelon؛ ساختار جزئی Node ۱۵
- شکل ۱۱-۱ شبیه ساز OverMelon؛ ساختار جزئی Link ۱۶
- شکل ۱۲-۱ شبیه ساز OverMelon؛ ساختار جزئی Traffic ۱۷
- شکل ۱۳-۱ شبیه ساز OverMelon؛ ساختار جزئی Time Line ۱۸
- شکل ۱۴-۱ شبیه ساز OverMelon؛ نقش Sequencer به عنوان لایه میانی Node و Time Line ۱۸
- شکل ۱۵-۱ شبیه ساز OverMelon؛ ساختار جزئی Sequencer ۱۹
- شکل ۱۶-۱ شبیه ساز OverMelon؛ نمای تقریبی قسمت ناظر ۱۹
- شکل ۱۷-۱ شبیه ساز OverMelon؛ ساختار جزئی OverMelon Engine ۲۰
- شکل ۱۸-۱ شبیه ساز OverMelon؛ نمای کلی موقعیت Handler ۲۰
- شکل ۱۹-۱ شبیه ساز OverMelon؛ ساختار جزئی GUI ۲۱
- شکل ۲۰-۱ شبیه ساز OverMelon؛ نمای کلی واسط کاربری ۲۲
- شکل ۲۱-۱ شبیه ساز OverMelon؛ نوار عنوان در واسط کاربری ۲۲
- شکل ۲۲-۱ شبیه ساز OverMelon؛ پانل Monitor در واسط کاربری ۲۳
- شکل ۲۳-۱ شبیه ساز OverMelon؛ پانل Control در واسط کاربری ۲۴
- شکل ۲۴-۱ شبیه ساز OverMelon؛ پانل Log Presenter در واسط کاربری ۲۶
- شکل ۲۵-۱ شبیه ساز OverMelon؛ پانل Property در واسط کاربری ۲۸
- شکل ۲۶-۱ شبیه ساز OverMelon؛ نوار وضعیت در واسط کاربری ۲۹

۱. محیط شبیه سازی

در حین توسعه الگوریتم های مربوط به شبکه نیاز به آزمودن آن ها برای اطمینان از صحت عملکردشان به شدت احساس می شود. با پیچیده شدن الگوریتم ها این نیاز به صورت پررنگ تری درک می شود. پیش از آنکه الگوریتم نهایی شود باید آزمایش های مختلفی بر روی آن انجام گیرد. این آزمایش ها می تواند در قالب پیاده سازی نرم افزاری یا سخت افزاری روش مورد تحقیق انجام شود. با توجه به هزینه بر بودن آزمایش های عملی، تست های نرم افزاری با فراهم آوردن محیط های شبیه سازی شده اهمیت بالایی دارد. در واقع در شبیه سازی تلاش می شود اثر پارامتر های کلیدی بر روی الگوریتم بررسی شود.

محیط های شبیه سازی فراوانی در حوزه شبکه های کامپیوتری وجود دارد که NS [1] یک نمونه از رایج ترین آن ها است. NS^۱ یک شبیه ساز شبکه است که رویداد ها را به صورت گسسته^۲ پردازش می

¹ Network Simulator

² Discrete Event

کند. این شبیه ساز در پروتکل های مسیریابی استفاده می شود و در تحقیقات مربوط به شبکه های اقتضایی¹ خیلی زیاد به کار می رود. NS پروتکل های معروف شبکه را پشتیبانی می کند و در حوزه متن باز شناخته شده است. با این حال مدلسازی در آن بسیار پیچیده و وقت گیر است. با توجه به این که واسط کاربر گرافیکی ندارد و کار با آن مستلزم یادگیری زبان اسکریپتی مخصوص به خود است و همین طور نیازمند آشنایی با مفاهیم تئوری صف و تکنیک های مدلسازی می باشد شبیه سازی با آن برای تازه کارها دشوار است.

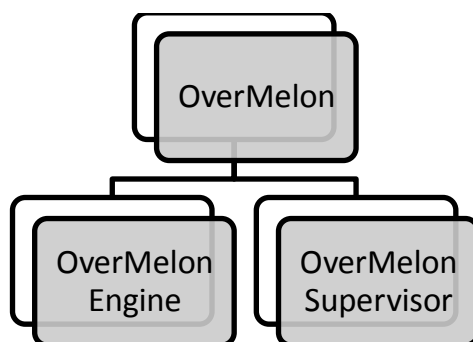
NS با زبان C++ نوشته شده و واسط شبیه ساز آن از طریق OTcl محیا شده است. کاربر پیکربندی شبکه را با نوشتن کد های OTcl توصیف می کند و برنامه اصلی NS، پیکربندی توصیف شده را شبیه سازی می کند.

با توجه به کاربردهای وسیع طراحی های مربوط به لایه کاربردی از لایه های شبکه، شبیه ساز های ویژه ای مربوط روی لایه پیاده شده است و مورد استفاده قرار می گیرد. به عنوان مثال OverSim یکی از آن ها به شمار می شود. OverSim [2] یک نرم افزار متن باز مخصوص کاربرد های روی لایه و نظیر به نظیر است. این نرم افزار توسط موسسه تلمتیکس در موسسه فن آوری Karlsruhe توسعه یافته و از زیرمجموعه های پروژه ScaleNet به شمار می رود. در واقع OverSim یک قالب شبیه سازی برای محیط های OMNeT++ به حساب می آید. این شبیه ساز مدل های متنوعی برای سیستم های ساختار یافته (مانند Chord، Kademlia و Pastry) و غیر آن (مانند GIA) دارد. از ویژگی های آن انعطاف پذیری بالا، واسط کاربر فعال، ارائه آمار، مودهای مسیرهای مختلف و امکان پیاده سازی در مقیاس بالا است.

¹ Ad-hoc

۱-۱. تعریف

جهت ارائه تعریف دقیق، OverMelon یک شبیه ساز شبکه های کامپیوتری در سطح روی لایه است. به عبارتی این شبیه ساز تنها به بررسی رفتار شبکه در لایه کاربردی می پردازد. به این ترتیب OverMelon نیازهای مربوط به شبیه سازی های روی لایه را برطرف می کند. این نرم افزار به زبان C# و در دو قسمت توسعه یافته. یک قسمت هسته اصلی نرم افزار است که امور مربوط شبکه روی لایه در آن انجام می شود و قسمت دیگر شامل واسط گرافیکی کاربر است. این قسمت ها در شکل ۱-۱ دیده می شود.



شکل ۱-۱ شبیه ساز OverMelon؛ نمای اجزای کلی

برای اطلاع بیشتر به وب سایت شبیه ساز [3] مراجعه کنید.

در ادامه به تشریح ساختار OverMelon می پردازیم.

۲-۱. ساختار

همان طور که در شکل ۱-۱ مشاهده می کنید OverMelon از دو قسمت مجزا تشکیل شده است:

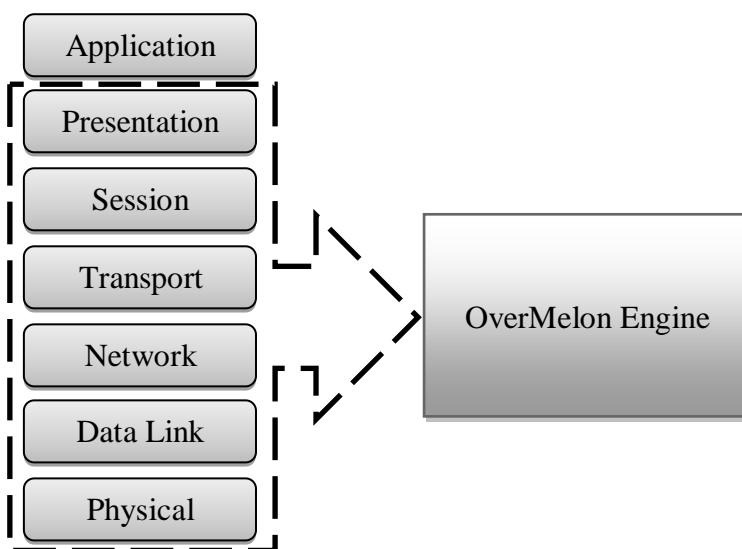
- هسته OverMelon
- ناظر OverMelon

دو قسمت هسته و ناظر از طریق یک واسط میانی با هم در ارتباط هستند. ابتدا به معرفی قسمت

هسته می پردازیم و سپس به سراغ ناظر می رویم.

۱-۲-۱. هسته OverMelon

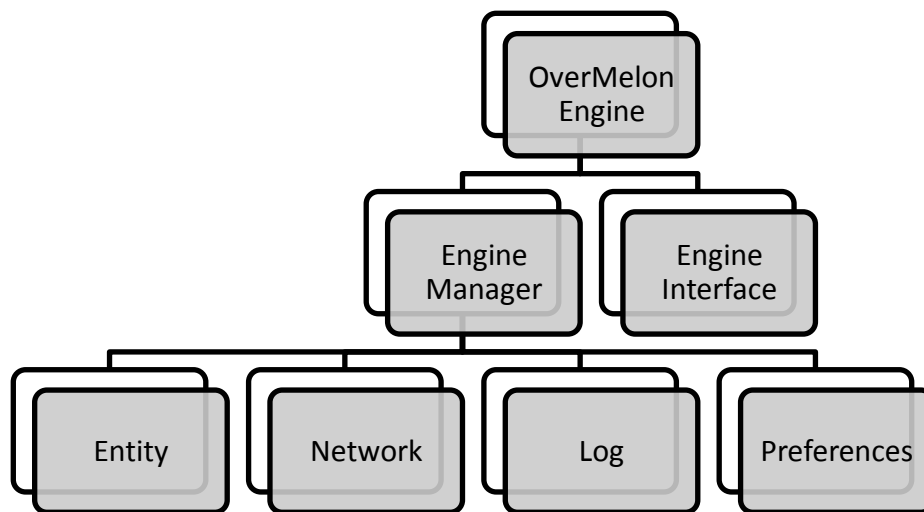
هسته قسمت اصلی نرم افزار OverMelon به حساب می آید. تمام کارهای مربوط به شبکه از جمله انتقال ترافیک بر عهده هسته است. هسته نرم افزار را می توان فارغ از ناظر اجرا کرده و از تمام قابلیت های آن استفاده کرد. برای روشن شدن مطلب لایه های OSI را مطابق شکل ۱-۲ در نظر بگیرید. همان طور که مشاهده می کنید هسته OverMelon در واقع نقش تمام لایه های پایین لایه کاربردی را بر عهده دارد.



شکل ۱-۲ مقایسه لایه های OSI با OverMelon

حال نگاه دقیق تری به ساختار هسته می کنیم. با مسامحه می توان این طور تصور کرد که هسته

OverMelon به صورت شکل ۱-۳ پایه ریزی شده است.



شکل ۳-۱ شبیه ساز OverMelon؛ ساختار جزئی

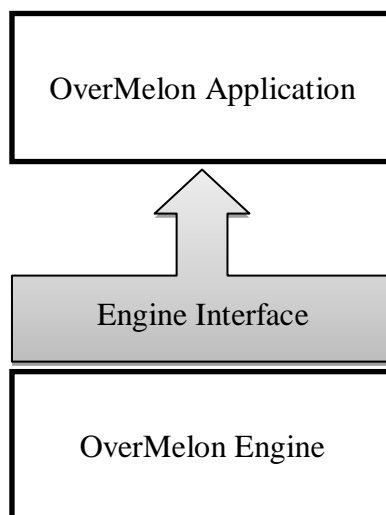
برای آشنایی بیشتر به بررسی مختصر هر یک از قسمت ها می پردازیم.

۱-۱-۲-۱. Engine Manager

همان طور که در شکل ۳-۱ مشاهده می کنید Engine Manager از پایه های هسته به شما می رود. زیر بخش های Entity، Network، Log و Preferences منجر به تشکیل Engine Manager می گردد. عملیات هماهنگی میان این زیر بخش ها در Manager انجام شده و این بخش ها از این طریق با هم در ارتباط هستند.

۲-۱-۲-۱. Engine Interface

یکی دیگر از قسمت های هسته زیر بخش Engine Interface است. این ماژول در حقیقت یک ماژول میانی است. همان طور که توضیح داده شد، هسته مسئولیت شش لایه ی پایینی OSI را بر عهده دارد و لایه کاربردی همان لایه بالایی OSI است. ارتباط میان لایه های پایینی و بالایی نیازمند واسطه هایی در پایین و بالا است. نقش این واسطه در لایه های پایینی بر عهده Engine Interface است. در واقع این ماژول پل ارتباطی میان هسته و لایه کاربردی است.

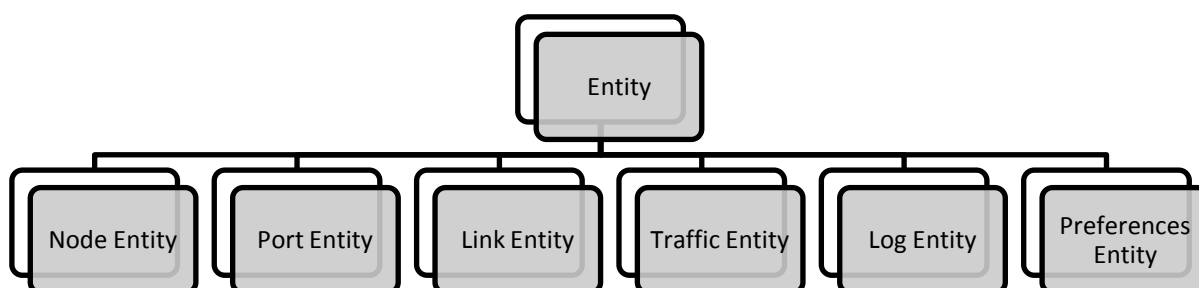


شکل ۴-۱ شبیه ساز OverMelon: نمای ساختار Engine Interface

ساختار عملکرد Engine Interface در شکل ۴-۱ مشاهده می شود.

Entity ۳-۱-۲-۱

زیر بخش Entity محتوی ماژول های پایه شبکه است. ساختار آن را در شکل ۵-۱ مشاهده می کنید. ماژول های Entity در واقع نگهدارنده اطلاعات هر جز هستند و هوشمندی ذاتی از خود ندارند. به عنوان مثال هر نمونه ی ماژول Node Entity اطلاعات مربوط به یک گره در شبکه را بر عهده دارد. می توان اینگونه بیان کرد که هر نمونه از ماژول های Entity معرف یک نمونه از آن نوع ماژول است. مثلا یک نمونه از ماژول Node Entity مبین گره موجود در شبکه است.



شکل ۵-۱ شبیه ساز OverMelon: ساختار جزئی Entity

Node Entity ۲-۳-۱-۲-۱

ماژول Node Entity شامل کلیه اطلاعاتی است که گره را تعریف می کند. به عنوان مثال محل گره،

وضعیت آن و درگاه های گره در این ماژول ذخیره می شود.

Port Entity ۳-۳-۱-۲-۱

هر گره یک سری درگاه ورودی و یک سری درگاه خروجی دارد. این درگاه ها برای اتصال به لینک ها به کار می روند. در واقع برای این که یک لینک به گرهی وصل شود باید به درگاه مربوط به آن گره متصل شود. ارتباط میان لینک و گره از طریق آن درگاه صورت می پذیرد. این درگاه توسط ماژول Port Entity تعریف می شود.

Link Entity ۴-۳-۱-۲-۱

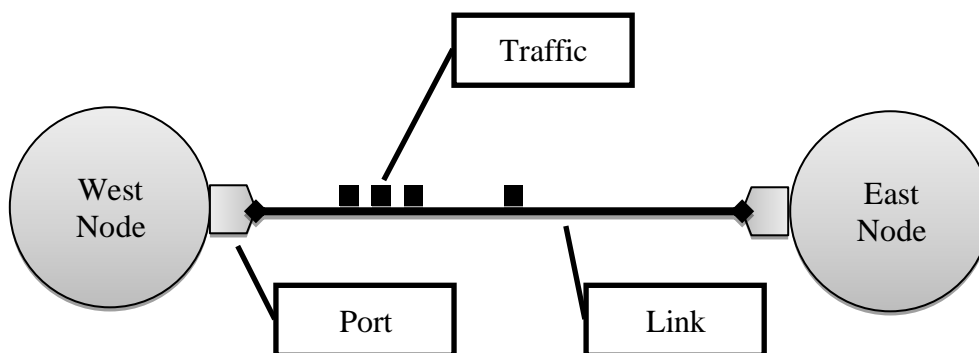
اتصال میان گره ها با کمک لینک ها انجام می شود. تاخیر بین دو گره از مشخصه های لینک میان آن دو است. هر یک از دو سر یک لینک به پورت مربوطه در گره ها وصل می شود و به این ترتیب ارتباط میان دو گره برقرار می شود. کنترل ترافیک روی شبکه بر عهده لینک است. لینک ها از موقعیت ترافیک هایی که بر روی هریک قرار دارد با خبر اند و هر لینکی ترافیک وارد شده را تا گره مقصد همراهی می کند و تحویل آن می دهد.

Traffic Entity ۵-۳-۱-۲-۱

اطلاعاتی که میان گره ها منتقل می شود همه در Traffic Entity خلاصه می گردد. هر ترافیک حاوی اطلاعات مشخصه گره های مبدا و مقصد است و پیام گره مبدا را نیز در بر می گیرد. مسئولیت کنترل ترافیک ها بر عهده لینک ها است.

برای این که با پروسه مربوط به انتقال پیام در OverMelon کمی آشنا شویم مثال ساده ای را بررسی می کنیم. در شکل ۱-۶ ارتباط کلی میان ماژول های Entity مشاهده می شود. فرض کنید گره غربی چهار بسته ترافیکی به گره شرقی ارسال کرده است. اتصال گره غربی با لینک میانی از طریق درگاه گره

برقرار می شود. وظیفه درگاه تنها اتصال گره و لینک متناظر با آن است. درگاه ها واسطه ای میان لینک ها و گره ها هستند. ترافیک ها بر روی لینک ها حرکت می کنند. لینک از لحظه ای که ترافیک را از درگاه گره تحویل می گیرد تا وقتی که آن را به درگاه گره مقصد تحویل می دهد، تنها مسئول ترافیک است. وقتی ترافیک به مقصد می رسد، تحویل درگاه گره شرقی می شود و از آنجا گره شرقی پردازش های لازم را بر روی ترافیک دریافتی انجام می دهد.



شکل ۱-۶ شبیه ساز OverMelon: نمای کلی ارتباط میان ماژول های Entity

Log Entity ۶-۳-۱-۲-۱

این ماژول یک ماژول غیر مرتبط به شبکه است. در واقع این ماژول یک ابزار کمکی شبیه ساز است که از آن برای ثبت اطلاعات استفاده می شود. اطلاعات مربوط به دریافت و ارسال ترافیک، ایجاد و حذف شدن گره ها و اطلاعات ثبتی لایه کاربردی از این جمله اند. این اطلاعات هم در بانک داده MySQL ذخیره می شود و هم در صفحه نمایش قسمت ناظر می تواند نمایش داده شود.

Preferences Entity ۷-۳-۱-۲-۱

برای اجرای هسته به یک سری اطلاعات پیش فرض نیاز داریم. به عنوان مثال برای این که بتوان داده های ماژول Log را بر روی بانک داده MySQL ذخیره کرد باید اطلاعات server، کلمه عبور و رمز مربوطه را داشته باشیم. این قبیل اطلاعات پیش فرض بر روی فایل XML همراه با نرم افزار شبیه ساز به نام preferences.xml ذخیره می شود و هنگام اجرای نرم افزار، این اطلاعات بازیابی شده و مورد

استفاده برنامه قرار می گیرد. ماژول Preferences Entity حاوی این اطلاعات است. در واقع هرگونه اطلاعاتی که از فایل Preferences.xml بازیابی می شود، در ماژول Preferences ذخیره می شود.

Log .۴-۱-۲-۱

زیر بخش Log وظیفه ثبت وقایع روی داده در شبکه را بر عهده دارد. اتفاق هایی که در شبکه روی می دهد به اطلاع این قسمت می رسد و توسط آن ثبت می شود. این وقایع شامل پنج مورد زیر است:

- ایجاد گره
- حذف گره
- ارسال ترافیک توسط گره
- دریافت ترافیک توسط گره
- پیغام های مربوط به لایه کاربردی

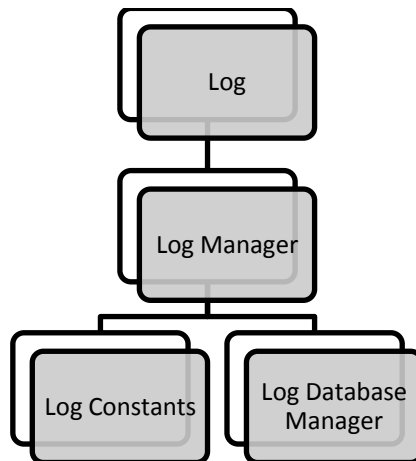
در مورد پیغام های مربوط به لایه کاربردی توضیح بیشتر مفید به نظر می رسد. توسعه دهنده لایه کاربردی بنا به نیاز می تواند در مواقعی که برایش مهم است برخی پیغام ها را در ماژول Log ثبت کند تا از آن برای آمارگیری و بررسی رفتار شبکه استفاده کند. این پیغام ها تحت عنوان پیغام های مربوط به لایه کاربردی در ماژول Log ثبت می شود.

ماژول Log پیغام های دریافتی را به دو صورت ثبت می کند:

- پیغام ها در بانک داده MySQL ذخیره می شود.
- یک رونوشت از پیغام به قسمت ناظر ارسال می شود تا در صورت تمایل پیغام را در صفحه نمایش چاپ کند.

زیر بخش Log به صورت ساختار شکل ۷-۱ است. همان طور که مشاهده می کنید این زیر بخش از

سه قسمت Log Manager، Log Constants و Log Database Manager تشکیل شده است. به بررسی جزئیات مربوط به هر کدام از قسمت ها نمی پردازیم. تنها به این نکته بسنده می کنیم که مقادیر پیش فرض و ثابت به کار رفته در این قسمت در ماژول Log Constants ذخیره می شود و وظیفه ذخیره سازی پیغام ها در بانک داده بر عهده Log Database Manager است.



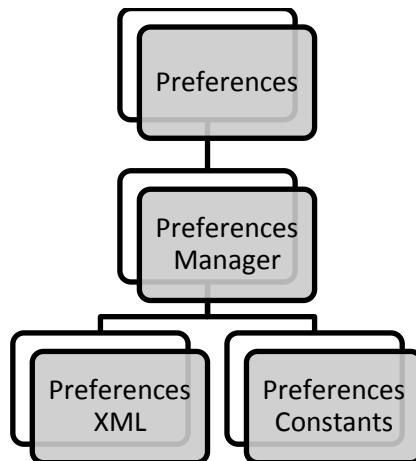
شکل ۱-۷ شبیه ساز OverMelon؛ ساختار جزئی Log

۱-۲-۱-۵. Preferences

برای اجرای هسته به یک سری اطلاعات پیش فرض نیاز داریم. به عنوان مثال برای این که بتوان داده های ماژول Log را بر روی بانک داده MySQL ذخیره کرد باید اطلاعات server، کلمه عبور و رمز مربوطه را داشته باشیم. این قبیل اطلاعات پیش فرض بر روی فایل XML همراه با نرم افزار شبیه ساز به نام preferences.xml ذخیره می شود و هنگام اجرای نرم افزار، این اطلاعات بازیابی شده و مورد استفاده برنامه قرار می گیرد. قسمت Preferences در ابتدای اجرای هسته، فایل Preferences.xml را کاوش کرده و اطلاعات ضروری برای فعالیت هسته را استخراج می کند. سپس این اطلاعات را در قالب Preferences Entity ذخیره می کند و در اختیار ماژول های مربوطه قرار می دهد.

زیر بخش Preferences به صورت ساختار شکل ۱-۸ است. همان طور که مشاهده می کنید این زیر بخش از سه قسمت Preferences Manager، Preferences Constants و Preferences XML تشکیل

شده است. به بررسی جزئیات مربوط به هر کدام از قسمت ها نمی پردازیم. تنها به این نکته بسنده می کنیم که مقادیر پیش فرض و ثابت به کار رفته در این قسمت در ماژول Preferences Constants ذخیره می شود و اطلاعات مربوط به نرم افزار در Preferences XML ذخیره می شود.

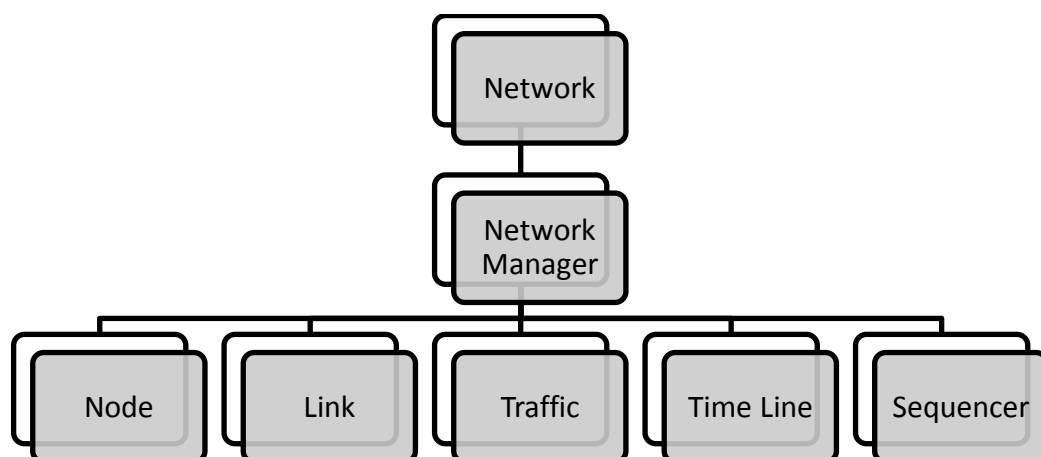


شکل ۸-۱ شبیه ساز OverMelon؛ ساختار جزئی Preferences

۶-۱-۲-۱ Network

تمام فعالیت های شبکه تحت نظر زیر بخش Network انجام می شود. می توان گفت که واحد مرکزی هسته در واقع زیر بخش Network است.

زیر بخش Network به صورت ساختار شکل ۹-۱ است. همان طور که مشاهده می کنید هر یک از قسمت های Node، Link، Traffic، Time Line و Sequencer مسئولیت بخشی از وظایف شبکه را بر عهده دارند که در مورد هر یک جداگانه توضیح داده می شود. سرگروه این قسمت Network Manager است که هماهنگ کننده این قسمت هاست.

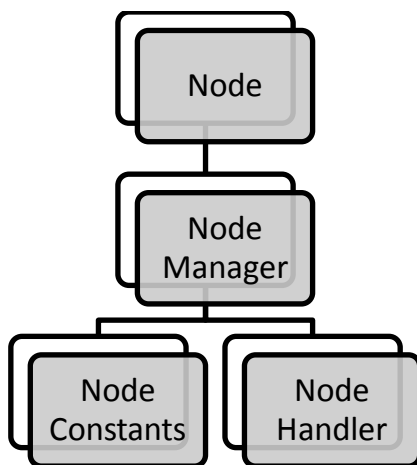


شکل ۱-۹ شبیه ساز OverMelon؛ ساختار جزئی Network

Node ۱-۶-۱-۲-۱

انجام وظایف یک گره در لایه هسته OverMelon بر عهده قسمت Node است. به عنوان مثال ارسال پیغام به درگاه و دریافت آن از درگاه، همین طور انجام پردازش های مربوطه همگی بر عهده این قسمت است.

زیر بخش Node به صورت ساختار شکل ۱-۱۰ است. همان طور که مشاهده می کنید این زیر بخش از سه قسمت Node Manager، Node Constants و Node Handler تشکیل شده است. به بررسی جزئیات مربوط به هر کدام از قسمت ها نمی پردازیم. تنها به این نکته بسنده می کنیم که مقادیر پیش فرض و ثابت به کار رفته در این قسمت در ماژول Node Constants ذخیره می شود و پردازش های مربوط به قسمت Node تماماً در Node Handler انجام می شود. سرگروه این قسمت Node Manager است که هماهنگ کننده این ماژول هاست.

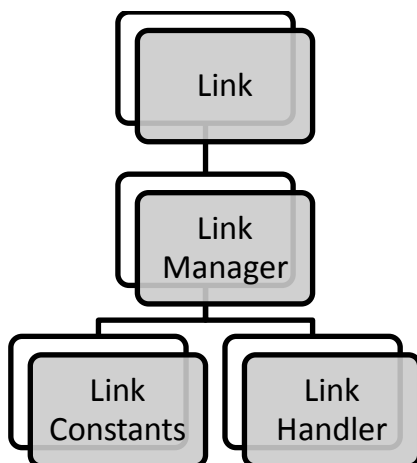


شکل ۱-۱۰ شبیه ساز OverMelon؛ ساختار جزئی Node

Link ۲-۶-۱-۲-۱

انجام وظایف یک لینک در هسته OverMelon بر عهده قسمت Link است. به عنوان مثال دریافت ترافیک از گره مبدا و تحویل آن به گره مقصد بر عهده این قسمت است.

زیر بخش Link به صورت ساختار شکل ۱-۱۱ است. همان طور که مشاهده می کنید این زیر بخش از سه قسمت Link Manager، Link Constants و Link Handler تشکیل شده است. به بررسی جزئیات مربوط به هر کدام از قسمت ها نمی پردازیم. تنها به این نکته بسنده می کنیم که مقادیر پیش فرض و ثابت به کار رفته در این قسمت در ماژول Link Constants ذخیره می شود و پردازش های مربوط به قسمت Link تماماً در Link Handler انجام می شود. سرگروه این قسمت Link Manager است که هماهنگ کننده این ماژول هاست.

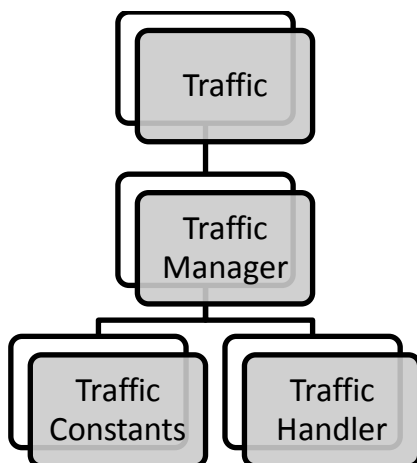


شکل ۱-۱۱ شبیه ساز OverMelon؛ ساختار جزئی Link

۱-۲-۱-۶-۳ Traffic

برخی کار های مربوط به ترافیک مانند تعیین محل ترافیک در لینک در این قسمت انجام می شود. البته زیر بخش Traffic با تعامل با زیر بخش Link محاسبات مربوطه را انجام می دهد و محل ترافیک را تعیین می کند.

زیر بخش Traffic به صورت ساختار شکل ۱-۱۲ است. همان طور که مشاهده می کنید این زیر بخش از سه قسمت Traffic Manager، Traffic Constants و Traffic Handler تشکیل شده است. به بررسی جزئیات مربوط به هر کدام از قسمت ها نمی پردازیم. تنها به این نکته بسنده می کنیم که مقادیر پیش فرض و ثابت به کار رفته در این قسمت در ماژول Traffic Constants ذخیره می شود و پردازش های مربوط به قسمت Traffic تماما در Traffic Handler انجام می شود. سرگروه این قسمت Traffic Manager است که هماهنگ کننده این ماژول هاست.

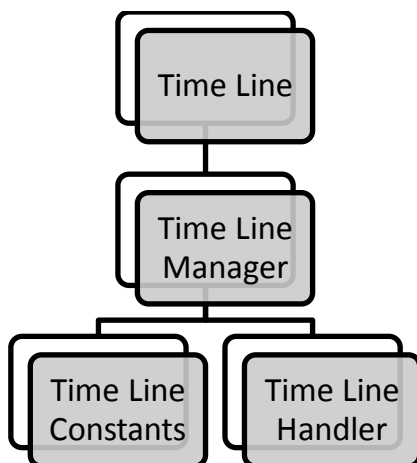


شکل ۱-۱۲ شبیه ساز OverMelon؛ ساختار جزئی Traffic

Time Line ۴-۶-۱-۲-۱

در یک محیط شبیه سازی وجود معیار زمانی برای انجام پردازش ها در مجموعه ی محیط ضروری است. زیر بخش Time Line چنین معیاری را فراهم می کند. Time Line در واقع زمان سنج محیط شبیه سازی است که هر یک از ماژول ها و قسمت های نرم افزار با مراجعه به آن خود را با زمان محیط هماهنگ می کنند و به انجام پردازش های مربوطه می پردازند. از قابلیت های یک زمان سنج امکان افزایش و کاهش دقت زمانی است، به طوری که زمان سریع تغییر کند یا به کندی پیش رود. این قابلیت توسط پارامتر Zoom در زیر بخش Time Line انجام می شود. در واقع پارامتر Zoom تعیین می کند که زمان با چه سرعتی سپری شود.

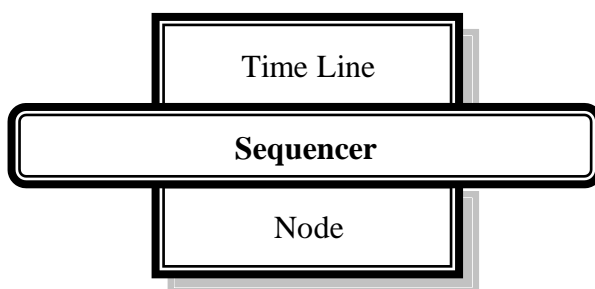
زیر بخش Time Line به صورت ساختار شکل ۱-۱۳ است. همان طور که مشاهده می کنید این زیر بخش از سه قسمت Time Line Manager، Time Line Constants و Time Line Handler تشکیل شده است. به بررسی جزئیات مربوط به هر کدام از قسمت ها نمی پردازیم. تنها به این نکته بسنده می کنیم که مقادیر پیش فرض و ثابت به کار رفته در این قسمت در ماژول Time Line Constants ذخیره می شود و پردازش های مربوط به قسمت Time Line تماماً در Time Line Handler انجام می شود. سرگروه این قسمت Time Line Manager است که هماهنگ کننده این ماژول هاست.



شکل ۱۳-۱ شبیه ساز OverMelon؛ ساختار جزئی Time Line

۱-۲-۱-۶-۵ Sequencer

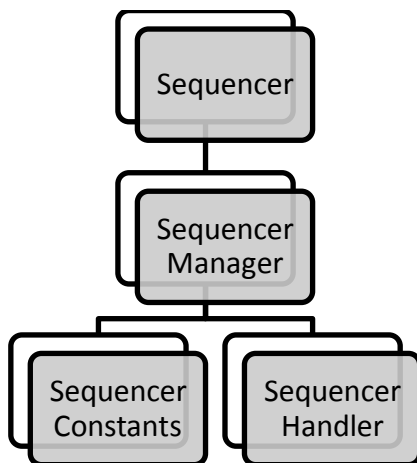
گره های به کار رفته در محیط شبیه سازی هر یک با سرعتی به پردازش ترافیک های ارسالی و دریافتی می پردازند. سرعت گره ها در پردازش یکی از معیار های موثر در این محیط شبیه سازی است. این سرعت به قابلیت های پایه سخت افزاری که نرم افزار OverMelon بر روی آن اجرا می شود بستگی دارد. برای کنترل سرعت پردازش گره های محیط شبیه سازی زیر بخش Sequencer به عنوان یک لایه میانی عمل می کند. این نقش در شکل ۱۴-۱ نمایش داده می شود.



شکل ۱۴-۱ شبیه ساز OverMelon؛ نقش Sequencer به عنوان لایه میانی Time Line و Node

زیر بخش Sequencer به صورت ساختار شکل ۱۵-۱ است. همان طور که مشاهده می کنید این زیر بخش از سه قسمت Sequencer Manager، Sequencer Constants و Sequencer Handler تشکیل شده است. به بررسی جزئیات مربوط به هر کدام از قسمت ها نمی پردازیم. تنها به این نکته بسنده می

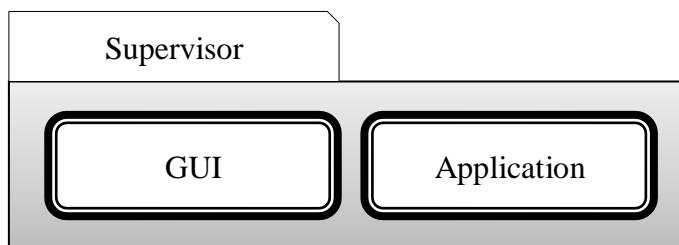
کنیم که مقادیر پیش فرض و ثابت به کار رفته در این قسمت در ماژول Sequencer Constants ذخیره می شود و پردازش های مربوط به قسمت Sequencer تماما در Sequencer Handler انجام می شود. سرگروه این قسمت Sequencer Manager است که هماهنگ کننده این ماژول هاست.



شکل ۱-۱۵ شبیه ساز OverMelon؛ ساختار جزئی Sequencer

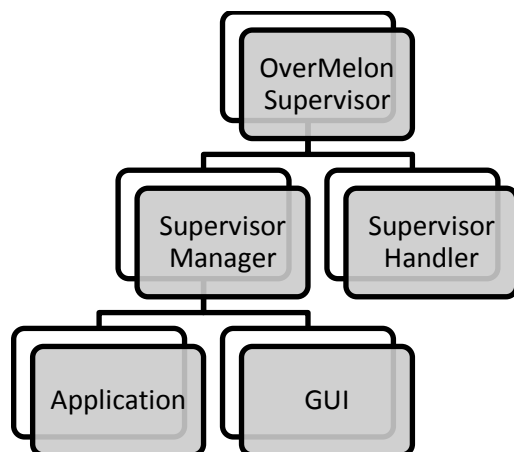
۲-۲-۱. ناظر

بر روی هسته OverMelon یک پوسته می کشیم و لایه کاربردی مدل OSI را بر روی آن پیاده می کنیم. پوسته ذکر شده همان قسمت ناظر است. در واقع ناظر تنها در برگزیده لایه کاربردی نیست بلکه شامل واسط های گرافیکی نرم افزار OverMelon نیز هست. شماتیک تقریبی آن در شکل ۱-۱۶ به تصویر کشیده شده است.



شکل ۱-۱۶ شبیه ساز OverMelon؛ نمای تقریبی قسمت ناظر

نگاه دقیق تری به ساختار ناظر می کنیم. با مسامحه می توان این طور تصور کرد که ناظر OverMelon به صورت شکل ۱-۱۷ پایه ریزی شده است.



شکل ۱۷-۱ شبیه ساز OverMelon؛ ساختار جزئی OverMelon Engine

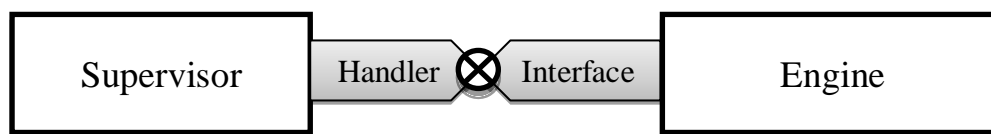
برای آشنایی بیشتر به بررسی مختصر هر یک از قسمت ها می پردازیم.

۲-۲-۲-۱ Supervisor Manager

این قسمت مرکز مدیریت ناظر است. تمام فعالیت هایی که در ناظر صورت می پذیرد رشته ای در Manager دارد. این قسمت به مدیریت لایه کاربردی می پردازد و در عین حال هماهنگی قسمت های مختلف واسط کاربر بر عهده آن است.

۳-۲-۲-۱ Supervisor Handler

برای ارتباط ناظر با هسته به واسطه هایی در هر دو طرف نیازمندیم. وظیفه واسطه در طرف هسته بر عهده Engine Interface است و نظیر آن در قسمت ناظر، Supervisor Handler است. واسط کاربری نرم افزار و همین طور پروتکل توسعه یافته لایه کاربردی به کرات نیاز دارند که با هسته شبیه ساز تعامل کنند. کانال این تعاملات در طرف ناظر، Handler است. نمای کلی موقعیت Handler در شکل ۱۸-۱ نشان داده شده است.



شکل ۱۸-۱ شبیه ساز OverMelon؛ نمای کلی موقعیت Handler

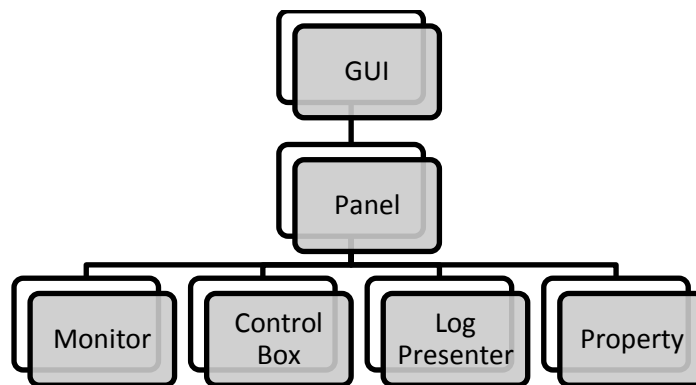
شکل ۱۸-۱ در واقع صورت کامل تر شکل ۴-۱ است.

Application .۴-۲-۲-۱

پیاده سازی پروتکل لایه کاربردی در این قسمت انجام می شود. همان طور که گفته شد تمام تعاملات مربوط به لایه کاربردی و Engine از طریق Handler صورت می گیرد.

GUI .۵-۲-۲-۱

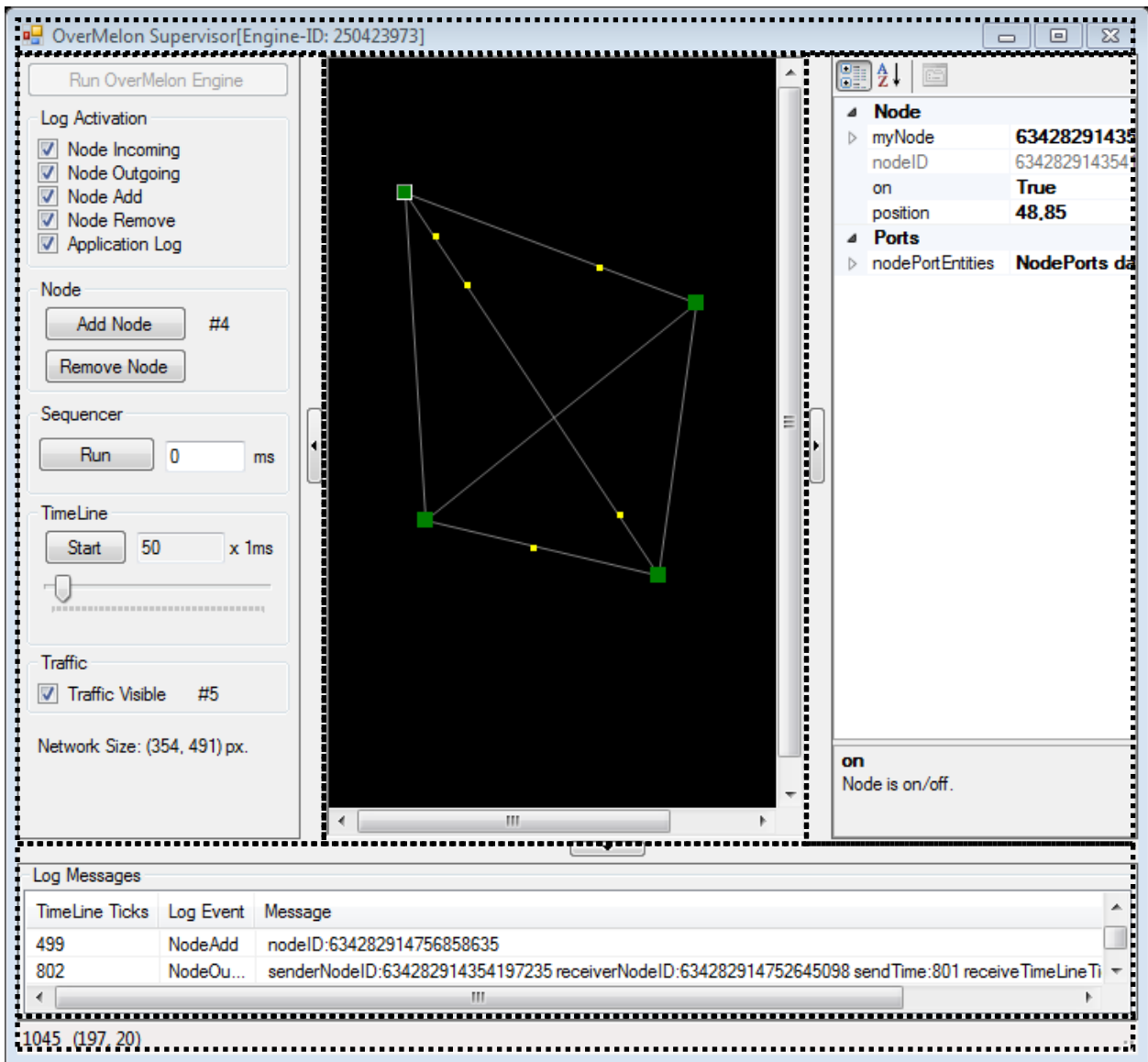
واسط کاربری نرم افزار شبیه ساز در قسمت Panel پیاده شده است. ساختار GUI در شکل ۱۹-۱ مشاهده می شود.



شکل ۱۹-۱ شبیه ساز OverMelon؛ ساختار جزئی GUI

GUI از یک پانل گرافیکی تشکیل شده است که تمام تعاملات شبیه ساز و کاربر از طریق آن صورت می گیرد. این پانل چهار زیر مجموعه فعال گرافیکی دارد که در چهار قسمت خود آن ها را جای داده و با فراهم آوردن یک محیط جذاب و آشنا برای کاربر مسیر های ساده ای برای دسترسی به امکانات OverMelon ایجاد کرده است. بسیاری از پیچیدگی های محیط هسته OverMelon را در خود هضم می کند و قابلیت های کلیدی را در اختیار کاربر می گذارد.

نمای کلی واسط کاربری را در شکل ۲۰-۱ می بینید.



شکل ۲۰-۱ شبیه ساز OverMelon؛ نمای کلی واسط کاربری

پانل واسط کاربری را به شش قسمت تقسیم کرده و در مورد هر کدام جداگانه توضیح می دهیم.

Title Bar ۱-۵-۲-۲-۱

در بالای پانل نوار عنوان قرار دارد. در این نوار مشخصه یکتای شناسایی هسته OverMelon پس از برقراری اتصال با هسته نمایش داده می شود. مثالی از مشخصه هسته را در شکل ۲۱-۱ ملاحظه می کنید.

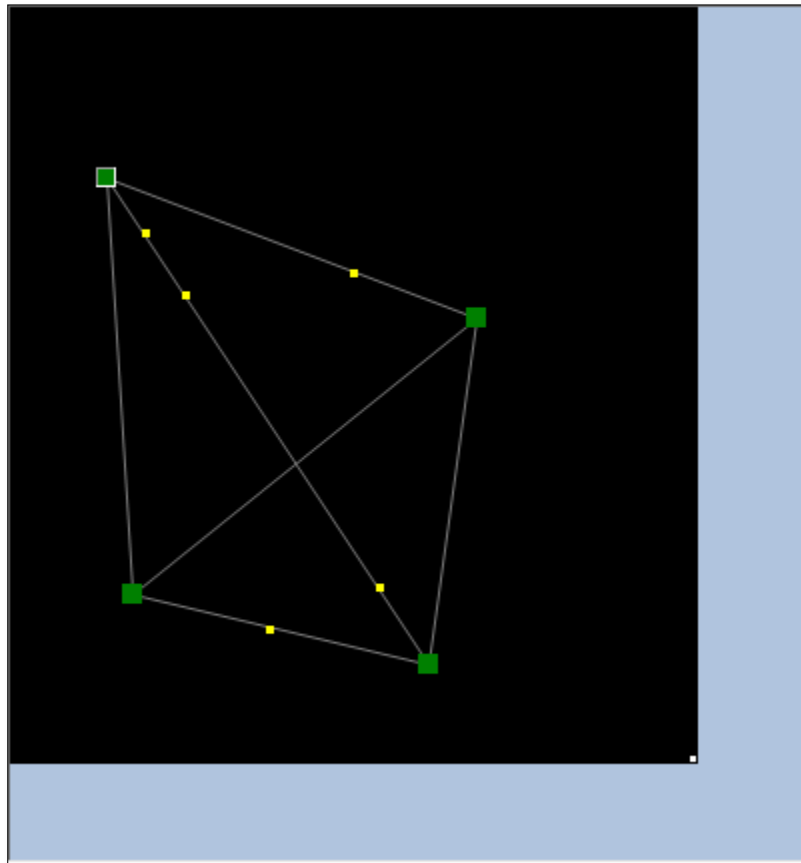


شکل ۲۱-۱ شبیه ساز OverMelon؛ نوار عنوان در واسط کاربری

Monitor ۲-۵-۲-۲-۱

پانل سیاه وسط واسط کاربر، فضای محیط شبیه ساز را نشان می دهد در این پانل گره ها قرار می گیرند و با لینک هایی به یکدیگر متصل می شوند. بسته های ترافیکی نیز بر روی لینک ها از یک گره به گره دیگر حرکت می کنند. با حرکت دادن مربع کوچک پایین سمت راست پانل می توانیم اندازه پانل را تغییر دهیم.

نمای کلی پانل Monitor در شکل ۲۲-۱ ملاحظه می شود. در این مثال چهار گره با رنگ سبز در پانل قرار دارند و پنج بسته ترافیکی با رنگ زرد میان گره ها جابجا می شود.



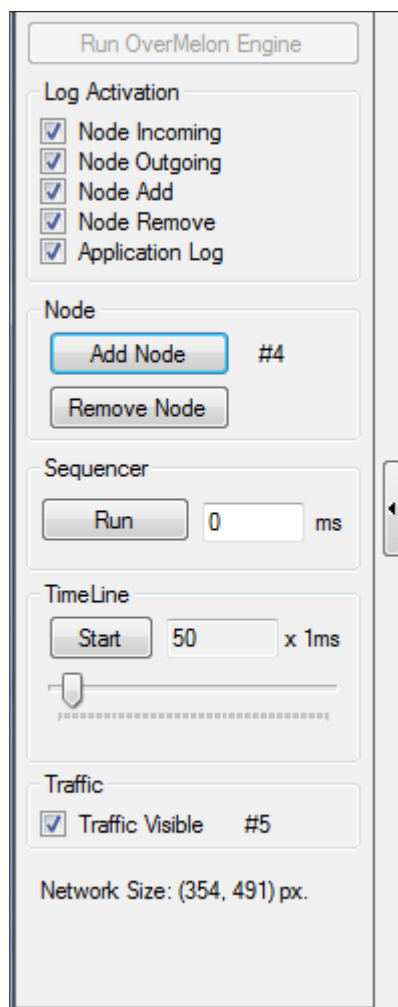
شکل ۲۲-۱ شبیه ساز OverMelon؛ پانل Monitor در واسط کاربری

Control Box ۳-۵-۲-۲-۱

ابزار های مدیریتی نرم افزار در پانل Control قرار دارد. با کلیک بر دکمه Run OverMelon

Engine، هسته شبیه ساز فعال شده و شماره مشخصه آن در نوار عنوان نوشته می شود. بدین ترتیب بقیه قسمت های پانل کنترل فعال می شوند.

نمای کلی پانل Control را در شکل ۱-۲۳ مشاهده می کنید. در صورتی که این پانل فضای زیادی از صفحه نمایش را اشغال می کند، با کلیک کردن بر روی دکمه مثلث می توان آن را به سمت چپ پنجره چسباند یا این که با قرار دادن ماوس بر روی خط باریک عمودی سمت راست پانل، اندازه پانل را تغییر داد.



شکل ۱-۲۳ شبیه ساز OverMelon؛ پانل Control در واسط کاربری

در قاب Log Activation با تیک کردن هر کدام از گزینه ها امکان ثبت وقایع روی داده در شبکه را فعال می کنید. به عنوان مثال اگر گزینه Node Incoming تیک خورده باشد، هر گره با دریافت ترافیک

از ماژول Log می خواهد تا رویداد دریافت ترافیک را ثبت کند. اگر این گزینه تیک نخورده باشد، چنین درخواستی به ماژول Log ارسال نمی شود. این قابلیت برای پنج نوع رویداد ممکن در ماژول Log فراهم شده است.

در قاب Node امکان اضافه کردن و یا حذف نمودن گره ها از شبکه وجود دارد. مثلاً برای اضافه کردن گره کافیست روی دکمه Add Node کلیک کنید تا فعال شود. سپس بر روی فضای سیاه رنگ پانل Monitor کلیک کنید تا دستور افزودن گره فعال گردد. به این ترتیب با وارد کردن مشخصات گره در پانل Property و کلیک بر روی دکمه Apply گره جدید به شبکه اضافه می شود.

در قاب Node تعداد گره های موجود در شبکه همواره نوشته می شود. در مثال شکل ۱-۲۳ همان طور که ملاحظه می کنید چهار گره به شبکه اضافه کرده ایم.

قاب Sequencer به کنترل قسمت Sequencer در هسته OverMelon می پردازد. سرعت پردازش گره ها را در کادر متنی در حسب میلی ثانیه وارد کنید و روی دکمه Run کلیک کنید. مثلاً اگر در کادر عدد ۵ را وارد کنیم و روی دکمه Run کلیک کنیم، هرکدام از گره ها در هر پنج میلی ثانیه یک بار مورد پردازش قرار می گیرند. اگر عدد صفر را در کادر وارد کنیم، با فشردن دکمه Run پردازش تنها برای یک بار انجام می شود و پس از آن متوقف می شود.

قاب Time Line به کنترل قسمت Time Line در هسته OverMelon می پردازد. با کلیک بر روی دکمه Start زمان در شبیه ساز سپری می شود. سرعت گذشت زمان در کادر متنی بر حسب میلی ثانیه نوشته می شود و با تغییر نوار غلطان تغییر می کند. در واقع با تغییر نوار غلطان پارامتر Zoom در هسته را تغییر می دهیم.

قاب Traffic به کنترل نمایش ترافیک موجود در شبکه اقدام می کند. با تیک کردن گزینه Traffic

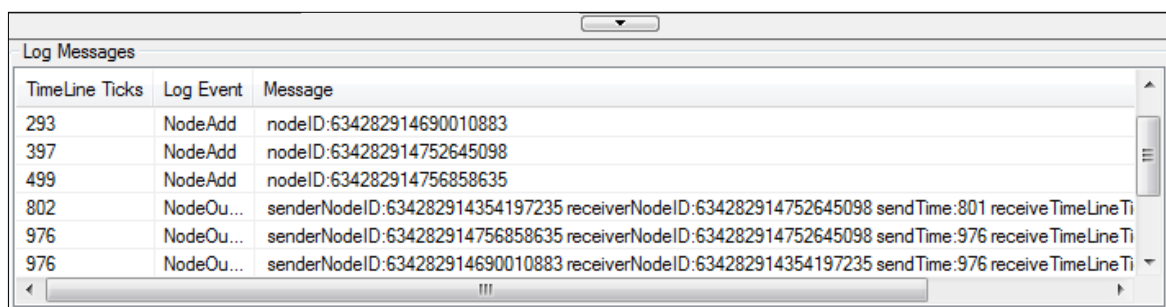
Visible بسته های ترافیکی زرد رنگ در شبکه نشان داده می شوند ولی به علت این که این کار مقدار زیادی از قدرت پردازشی پردازنده را می گیرد، می توان این گزینه را غیر فعال کرد. تعداد بسته های ترافیکی موجود در شبکه در قاب Traffic همواره نوشته می شود. در مثال شکل ۱-۲۳ همان طور که ملاحظه می کنید پنج بسته ترافیکی در شبکه موجود است.

ابعاد شبکه با عنوان Network Size بر حسب پیکسل در پنل Control مشاهده می شود. می توان ابعاد شبکه را در پنل Monitor تغییر داد.

۱-۲-۲-۵-۴ Log Presenter

نمایش رویداد های شبکه در پنل Log Presenter صورت می گیرد.

نمای کلی پنل Log Presenter را در شکل ۱-۲۴ مشاهده می کنید. در صورتی که این پنل فضای زیادی از صفحه نمایش را اشغال می کند، با کلیک کردن بر روی دکمه مثلث می توان آن را به پایین پنجره چسباند یا این که با قرار دادن ماوس بر روی خط باریک افقی بالای پنل، اندازه پنل را تغییر داد.



TimeLine Ticks	Log Event	Message
293	NodeAdd	nodeID:634282914690010883
397	NodeAdd	nodeID:634282914752645098
499	NodeAdd	nodeID:634282914756858635
802	NodeOu...	senderNodeID:634282914354197235 receiverNodeID:634282914752645098 sendTime:801 receive TimeLine Ti
976	NodeOu...	senderNodeID:634282914756858635 receiverNodeID:634282914752645098 sendTime:976 receive TimeLine Ti
976	NodeOu...	senderNodeID:634282914690010883 receiverNodeID:634282914354197235 sendTime:976 receive TimeLine Ti

شکل ۱-۲۴ شبیه ساز OverMelon: پنل Log Presenter در واسط کاربری

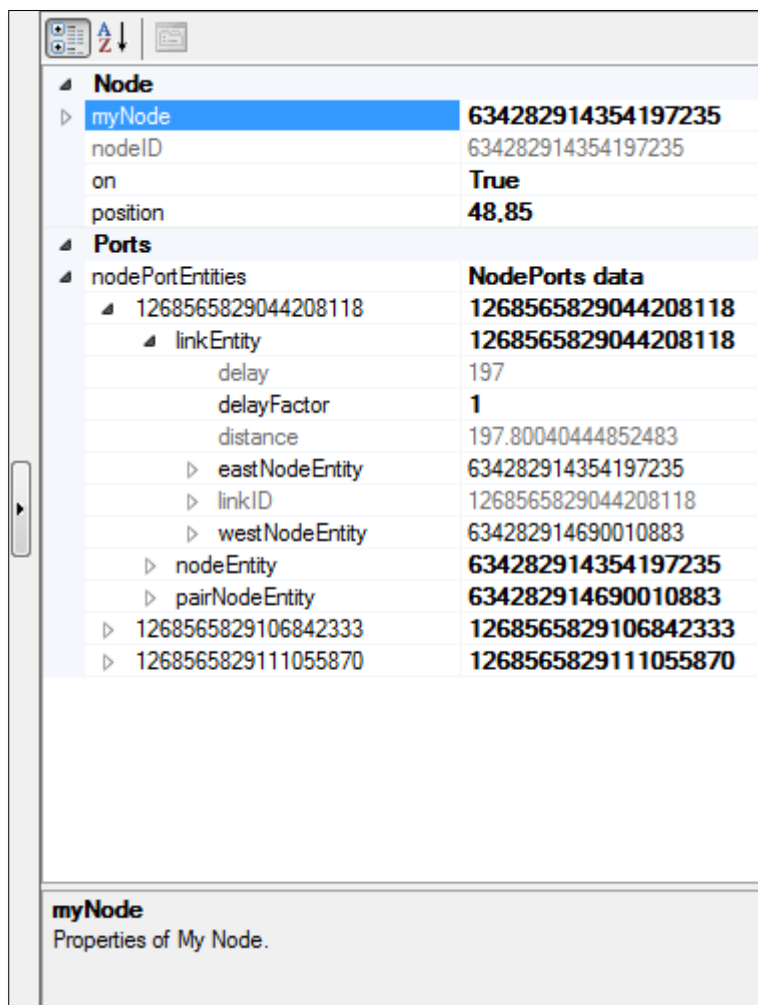
در این پنل در جدول Log Messages، زمان وقوع رویداد بر حسب مرجع Time Line آمده است. نوع رویداد ذیل عنوان Log Event مشخص می شود. توضیحات مربوط به رویداد نیز در فیلد Messages می آید. به عنوان مثال در شکل ۱-۲۴ در زمان ۲۹۳ بر حسب میلی ثانیه گره با شماره مشخصه 634282914690010883 به شبکه افزوده شده است. این اطلاعات در بانک داده MySQL نیز

ذخیره شده است.

Property ۵-۵-۲-۲-۱

بخش عمده ای از کار با نرم افزار به پانل Property معطوف می شود. در این پانل مشخصات گره ها، لینک ها و بسته های ترافیکی به نمایش در می آید. با انتخاب گره یا لینک یا بسته ترافیکی پانل Property مشخصات آن را نشان می دهد و می توانید مشخصات را به دلخواه تغییر دهید و مقادیر جدید را تنظیم کنید.

نمای کلی پانل Property را در شکل ۱-۲۵ مشاهده می کنید. در صورتی که این پانل فضای زیادی از صفحه نمایش را اشغال می کند، با کلیک کردن بر روی دکمه مثلث می توان آن را به سمت راست پنجره چسباند یا این که با قرار دادن ماوس بر روی خط باریک عمودی سمت چپ پانل، اندازه پانل را تغییر داد.



شکل ۱-۲۵ شبیه ساز OverMelon: پانل Property در واسط کاربری

در مثال شکل ۱-۲۵ مشخصات یک گره ملاحظه می شود. با تغییر مقدار مشخصه on به false گره انتخاب شده خاموش می شود و عملیات پردازشی در آن صورت نمی گیرد. محل گره در شبکه در مشخصه position آمده است که می توانیم آن را تغییر دهیم. همین طور درگاه های گره لیست شده اند و از آن طریق به لینک هایی که گره به آن های متصل است دسترسی داریم.

۱-۲-۲-۵-۶ Status Bar

در پایین پانل نوار وضعیت قرار دارد. در این نوار زمان شبیه ساز منطبق با Time Line بر حسب میلی ثانیه دیده می شود و موقعیت ماوس بر روی پانل Monitor بر حسب مختصات افقی و عمودی وجود دارد.

نمای کلی نوار وضعیت را در شکل ۲۶-۱ مشاهده می کنید.

1045 (114, 32)

شکل ۲۶-۱ شبیه ساز OverMelon: نوار وضعیت در واسط کاربری

در مثال شکل ۲۶-۱ زمان ۱۰۴۵ میلی ثانیه است و ماوس در مختصات ۱۱۴ افقی و ۳۲ عمودی قرار

دارد.

پیوست‌ها

پیوست ۱ – مثالی از اجرای محیط شبیه ساز

برای آشنایی با نحوه کار واسط کاربری به تشریح مثالی می پردازیم.

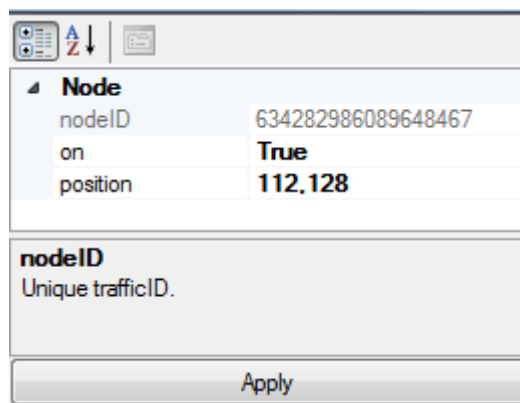
فرض کنید می خواهیم شبکه ای متشکل از سه گره را آزمایش کنیم. برای این منظور نرم افزار

OverMelon Supervisor را اجرا می کنیم و روی دکمه Run OverMelon Engine در پانل Control

کلیک می کنیم تا هسته شبیه ساز اجرا شود.

برای افزودن گره، بر روی دکمه Add Node کلیک کنید تا فعال شود. سپس بر روی صفحه مشکی

پانل Monitor کلیک کنید. با این کار پانل Property به صورت شکل پ-۱ به نمایش در می آید.



شکل پ-۱ شبیه ساز OverMelon؛ پانل Property هنگام افزودن گره

محل گره در پانل Monitor همان جایی است که کلیک کرده اید. می توانید آن را تغییر دهید و یا وضعیت گره را در حالت خاموش/روشن تنظیم کنید. پس از آن روی دکمه Apply کلیک کنید تا گره به شبکه افزوده شود. گره افزوده شده در حالت خاموش با مربع قرمز رنگ و در حالت روشن با مربع سبز رنگ در پانل Property به نمایش در می آید.

فرآیند افزودن گره جدید را دو بار دیگر تکرار کنید تا در کل سه گره در شبکه داشته باشیم. دقت کنید که گره ها روشن باشند (با رنگ سبز). با کلیک روی دکمه Add Node آن را غیر فعال کنید. سپس پردازنده گره ها را فعال می کنیم. در کادر متنی قاب Sequencer عدد یک را وارد کنید و روی دکمه Run کلیک کنید. با این کار دیگر امکان افزودن و یا حذف گره نداریم.

محیط شبیه سازی در زمان صفر قرار دارد. برای فعال کردن زمان سنج در قاب Time Line روی دکمه Start کلیک کنید. همان طور که در نوار وضعیت پایین صفحه ملاحظه می کنید، زمان با سرعت مشخص شده در قاب Time Line سپری می شود.

بر روی یکی از گره ها به دلخواه کلیک کنید. دور گره انتخاب شده کادر کشیده می شود و از سایر گره ها متمایز می شود. اگر می خواهید از گره انتخاب شده به گره دیگری پیغام آزمایشی بفرستید، کافیست همان طور که گره مبدا را انتخاب کرده اید با ماوس بر روی گره مقصد کلیک راست کنید. به

این ترتیب ترافیک حاوی پیغام آزمایشی از گره مبدا به سمت گره مقصد بر روی لینک بین دو گره حرکت می کند و در گره مقصد پردازش می شود.

نتیجه عمل را می توانید در جدول Log Messages از پانل Log Presenter ملاحظه کنید.

منابع و مراجع

- [1] Network Simulator. [Online]. http://nsnam.isi.edu/nsnam/index.php/Main_Page
- [2] OverSim. [Online]. <http://www.oversim.org/>
- [3] OverMelon. [Online]. <http://ee.sharif.edu/~overmelon/>