دانشکده مهندسی کامپیوتر

عنوان پروژه:

ایجاد یک شبکه توزیع در بستر پروتکل SIP جهت انتقال پیام

مستند نهایی (کلیه فازها) – سه شنبه ۱۳۹۵/۰۹/۰۲

دانشجویان:

مرتضی ذاکری - محسن امیریان – سعید امیری

استاد:

دکتر محسن شریفی

دستیار حل تمرین:

محمد مهدی طالبی

پاییز ۱۳۹۵

فهرست مطالب

۱ مقدمه ۱

۱ـ۱ توضیحات ضروری فاز اول ۱

۱ـ۲ توضیحات ضروری فاز دوم ۱

۱ـ۳ توضیحات ضروری فاز سوم ۲

۱ـ۴ توضیحات ضروری فاز چهارم ۲

۱ـ۵ تاریخچه تغییرات سند ۳

۱ـ۶ تقسیم وظایف ٤

۲ مفاهیم اولیه SIP ۴

۲ـ۱ عناصر شبکه توزیع ٥

۲ـ۲ پیام رسانی ٦

۳ معماری شبکه توزیع ارسال و دریافت پیام ۸

۳ـ۱ دیدگاه سطح مولفه سیستم ۸

۳ـ۲ پیاده سازی موجودیت SERVER ۹

۳ـ۳ پیاده سازی موجودیت CLIENT ۱۰

۳ـ۴ بهم بست SERVER ها در شبکه توزیع (توپولوژی شبکه) ۱۱

۳ـ۵ ارتباط CLIENT ها با SERVER ها ۱۲

۳ـ۶ سناریو و الگوریتم ها ۱۳

۳ـ۶ـ۱ سناریو ثبت نام Client ۱۳

۳ـ۶ـ۲ سناریو مسیر یابی (ارسال و دریافت پیام) ۱۴

۳ـ۶ـ۳ تحلیل زمانی الگوریتم مسیر یابی ۱۵

۳ـ۶ـ۴ سناریو افزودن Proxy Server ۱۵

۳ـ۷ مدل بانک اطلاعات برنامه ۱۷

۴ بررسی معیارها و اهداف سیستم های توزیعی ۱۹

۴ـ۱ بررسی جایگاه معماری ۲۰

۴ـ۲ قابلیت دستیابی ۲۱
۴ـ۳ شفافیت ۲۱
۴ـ۳ـ۱ شفافیت دستیابی ۲۱
۴ـ۳ـ۲ شفافیت مکانی ۲۲
۴ـ۳ـ۳ شفافیت مهاجرتی ۲۲
۴ـ۳ـ۴ شفافیت جابجایی ۲۲
۴ـ۳ـ۵ شفافیت در تکرار ۲۲
۴ـ۳ـ۶ شفافیت هم روندی ۲۳
۴ـ۳ـ۷ شفافیت خرابی ۲۳
۴ـ۴ باز بودن ۲۳
۴ـ۵ مقیاس پذیری ۲۳
۵ مشخصات فنی پروژه ۲۴
۶ منابع و ماخذ ۲۴

فهرست شکل ها

شکل ۱-۱. طرح واره ای از عملیات فاز دوم ۲
شکل ۱-۲. ساختار لایه ای پروتکل ها و جایگاه برنامه شبکه توزیع در آن..... ۵
شکل ۲-۲. نقش المان های مختلف در کنار یکدیگر برای شکل گیری یک ارتباط مبتنی بر SIP ۶
شکل ۳-۲. نمونه دیاگرام پیغام متنی..... ۷
شکل ۱-۳ نمودار مولفه ها (COMPONENT DIAGRAM) شبکه توزیع پیام ۹
شکل ۲-۳. نمودار کلاس لایه مدل SERVER در قالب UML ۱۰
شکل ۳-۳ نمودار کلاس لایه مدل سمتCLIENT در قالب UML..... ۱۱
شکل ۴-۳. ارتباطات مختلف در شبکه توزیع با توپولوژی RING-STAR ۱۳
شکل ۵-۳. سناریو ثبت نام ۱۴
شکل ۶-۳ طرح واره ای از نحوه مسیر یابی ۱۵
شکل ۷-۳. طرح واره ای از نحوه اتصال دو سرویس دهنده ۱۶
شکل ۸-۳. مراحل اضافه شدن گره جدید به انتهای شبکه RING-STAR ۱۷
شکل ۹-۳ گره اضافه شده به شبکه توزیع..... ۱۷
شکل ۱۰-۳. شمای پایگاه داده شبکه توزیع ۱۸
شکل ۱۱-۳. مدل داده شبکه توزیع در قالب نمودار ER ۱۹

فهرست جدول ها

جدول ۱-۱ شرح تغییرات سند..... ۳
جدول ۱-۲ تقسیم تخمینی وظایف..... ۴

۱ مقدمه

شبکه انتقال پیام یک برنامه کاربردی مبتنی بر ا ستاندارد SIP[۱] ا ست که تعدادی کاربر می توانند از طریق آن به در قالب پیام های متنی، با یکدیگر گفت و گو کنند. چالش اصـلی در طراحی و پیاده سازی چنین شبکه ای ایجاد یک **میان افزار توزیع شده** است که از طریق آن ارتباط بین Serverها با یکدیگر برقرار و کارکردهایی نظیر مسیر یابی و ارسال (Forward) پیام میسر می شود.

در این پروژه سعی شده ا ست ضمن برر سی جنبه های تئوری چنین سیستم توزیع شده ای، طراحی برای آن مطرح شود. در فاز نهایی، پیاده سازی ساده ای از آن چه در در فاز طراحی مد نظر بوده اسـت، ارایه شـده اسـت. در پایان نیز ویژگی ها مطرح در سـیسـتم های توزیعی نظر قابلیت دسـتیابی، شـفافیت، مقیاس پذیری و مدل های معماری، روی طرح پیشنهادی بررسی شده است. این پروژه به عنوان یک تمرین عملی انجام شده در درس سیستم های توزیعی می باشد. مطالبی که در ادامه عنوان شده اند، ممکن است نیازمند برخی اصلاحات باشند.

۱ـ۱ توضیحات ضروری فاز اول

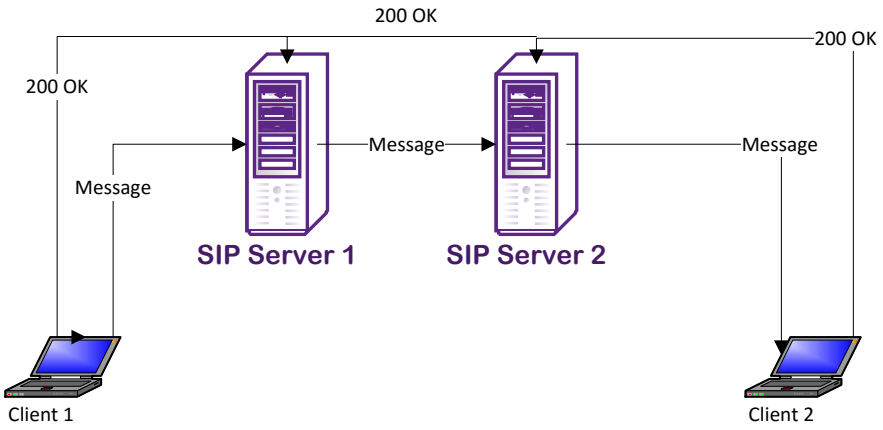
در فاز اول ارتباط دو client با استفاده از یک سرویس دهنده SIP برقرار شده است. در این سرور ابتدا Client ها خود را Register نموده و سپس به تبادل پیام می پردازند. با توجه به این که Dialog پیام متنی از سـناریو سـاده ای پیروی می کند، ما در این فاز سـناریو ایجاد یه ارتباط VoIP را بر بستر پروتکل UDP پیاده سازی کرده ایم، در واقع از پیام های Invite و Bye نیز بهره جسته ایم.

۱ـ۲ توضیحات ضروری فاز دوم

در فاز دوم ارتباط دو client با ا ستفاده از **دو سرویس دهنده SIP** برقرار شده ا ست. در هریک از این Serverها ابتدا Client ها خود را Register نموده و سپس به طور غیر مستقیم

[۱] Session Initial Protocol

به تبادل پیام می پردازند. در فاز اول، همان طور که در مستندات مربوط به آن فاز، ذکر گردید ما از سناریو ایجاد یک ارتباط VoIP بر بستر پروتکل UDP استفاده کردیم، حال آن که سناریو Instant Messaging (IM) یا همان *پیام رسانی* بسیار ساده تر است. در این فاز ما صرفا سناریو IM را پیاده کرده و قسمت های اضافی مربوط به سناریو VoIP را از کد برنامه حذف نموده ایم. در واقع پیام های Invite و Bye هم اینک حذف شده اند. سناریو نهایی پیاده سازی شده در این فاز، به شکل ۱-۱ است.



شکل ۱-۱. طرح واره ای از عملیات فاز دوم

۱ـ۳ توضیحات ضروری فاز سوم

در فاز سوم از پروژه به شرح معماری در نظر گرفته شده برای شبکه توزیع پرداخته ایم که در بخش سوم به صورت مستند وار مطرح شده است. بخش دوم به مفاهیم اولیه پروتکل SIP پرداخته و در بخش سوم، ابتدا طرح معماری و سپس روند اجرای سناریوهای مختلف به صورت الگوریتم وار ذکر شده است.

۱ـ۴ توضیحات ضروری فاز چهارم

در فاز چهارم و نهایی پروژه، بر اساس معماری پیشنهادی فاز سوم، شبکه توزیع ارسال و دریافت پیام پیاده سازی شده است. بدین ترتیب در پاین این فاز، یک شبکه توزیعی کامل

برای تبادل پیام خواهیم دا شت. سعی شده ا ست پیاده سازی کاملی از آنچه گفته شده انجام شود. این پیاده سازی به تدریج تکمیل می گردد.

۵ـ۱ تاریخچه تغییرات سند

سـند فوق در طول مراحل اجرای پروژه مورد بازبینی و ویرایش قرار گرفته و در هر فاز اقدامات انجام شده به آن اضافه شده است. محتوای سند به گونه ای تنظیم شده است که هم به عنوان یک گزارش از بالا به پایین قابل مطالعه باشـد و هم به عنوان ارجاع به یک بخش خاص. شرح تغییرات در جدول ۱-۱ ذکر شده است.

جدول ۱-۱ شرح تغییرات سند

ردیف	تاریخ	شرح تغییرات	افراد	نگارش
۱	۹۵ /۰ ۸ /۰۱	تدوین طرح اولیه (فاز دوم)	ذاکری – امیریان – امیری	۱,۰
۲	۹۵ /۰۸ /۰ ۹	تکمیل طرح – اضافه کردن سناریو ها (فاز سوم)	ذاکری – امیریان – امیری	۳,۱
۳	۹۵ /۰ ۸ /۱۰	اصلاح برخی موارد	ذاکری – امیریان – امیری	۴,۰
۴	۹۵ /۰۸ /۱۱	ارایه حضوری طرح	ذاکری	۴,۰
۵	۹۵ /۰۸ /۱۵	تکمیل طرح – اضافه کردن ویژگی های سیستم های توزیع شده	امیریان	۴,۱
۶	۹۵ /۰۸ /۱۹	تغییر توپولوژی شبکه و سناریو مسیر یابی	ذاکری – امیریان – امیری	۵,۰
۷	۹۵/۰۸/۲۸	افزودن مستندات فاز چهارم	ذاکری-امیریان - امیری	۶,۰

۶ـ۱ تقسیم وظایف

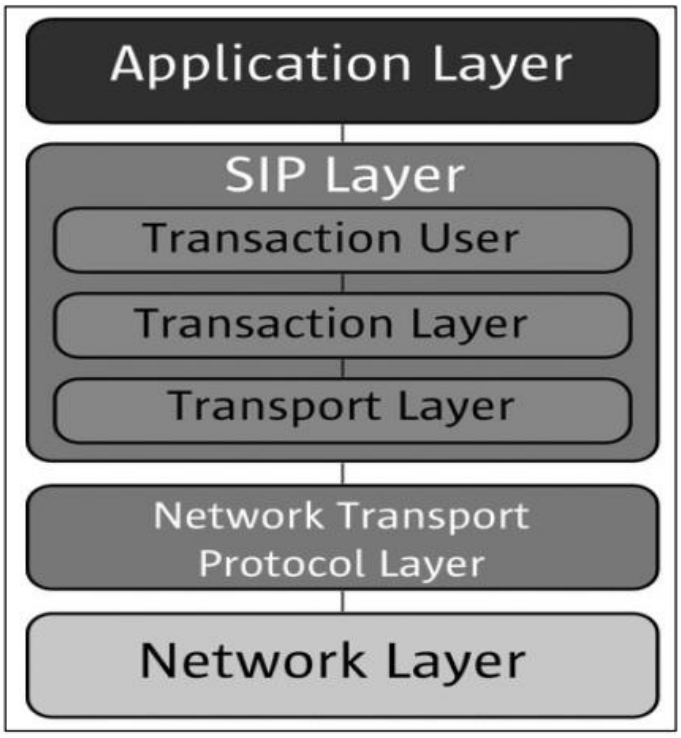
پروژه در قالب یک کارگروهی انجام شده است. در جدول زیر نحوه تقسیم وظایف بین اعضای گروه ذکر شده است.

جدول ۱-۲ تقسیم تخمینی وظایف

ردیف	شرح وظیفه	فرد مسئول
۱	مطالعه استاندارد SIP	ذاکری
۲	مطالعه و بررسی کتابخانه LumiSoft	امیریان
۳	پیاده سازی واسط گرافیکی پروژه (فاز اول)	ذاکری
۴	پیاده سازی فاز اول	امیریان - امیری
۵	پیاده سازی فاز دوم	امیریان – امیری - ذاکری
۶	فاز سوم (طراحی)	امیریان – امیری - ذاکری
۷	پیاده سازی فاز چهارم	امیری – امیریان
۸	پیاده سازی واسط گرافیکی پروژه (فاز نهایی)	امیری
۹	اصلاح مستندات (فاز نهایی)	ذاکری

۲ مفاهیم اولیه SIP

در این بخش برخی از مفاهیم اصلی و واژگان استفاده شده در پروژه توصیف شده است. ابتدا نگاهی به ساختار لایه ای برنامه در SIP و جایگاه آن در پشته پروتکلی شبکه می اندازیم و جایگاه برنامه خود را در آن نشان می دهیم. این ساختار در شکل ۱-۲ آمده است.



شکل ۱-۲. ساختار لایه ای پروتکل ها و جایگاه برنامه شبکه توزیع در آن

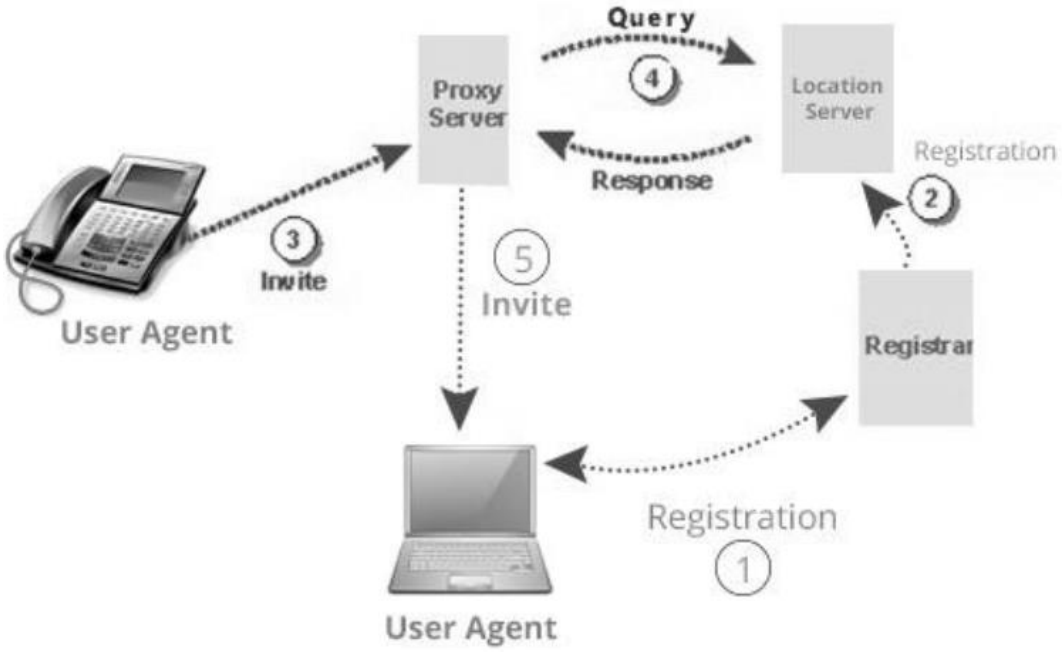
در این ساختار برنامه کاربردی در لایه کاربرد و بر بالای پروتکل SIP قرار دارد. در واقع ارتباط برقرار شده در شبکه توزیع توسط پروتکل SIP استاندارد می گردد.

۲ـ۱ عناصر شبکه توزیع

شبکه توزیع مطرح شده در این جا شامل عناصری است که در استاندارد SIP مطرح تا به شکل گیری شبکه مبتنی بر آن کمک کنند. در SIP هر عنصر شبکه با یک SIP URI شناسایی می شود که شبیه آدرس یا شماره شناسایی یکتاست. ما المان های زیر را در پروژه خود لحاظ کرده ایم.

۱. User Agent
۲. Proxy Server
۳. Registrar Server

وظایف المان های فوق در ا ستاندارد RFC 3261 شرح داده شده ا ست و در این جا از ذکر آن خودداری می کنیم. شکل زیر نقشی که هریک از عناصر نامبرده در شکل گیری یک ارتباط در شبکه توزیع ما را بازی می کند، به تصویر کشیده است:

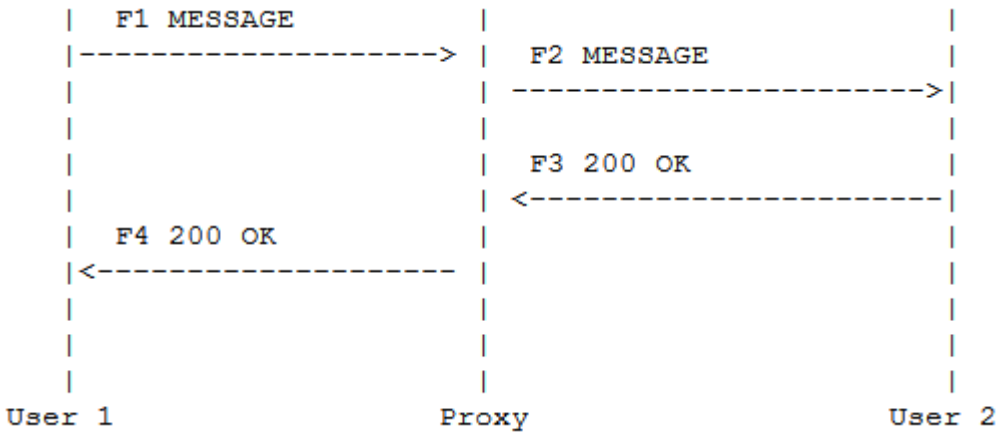


شکل ۲-۲. نقش المان های مختلف در کنار یکدیگر برای شکل گیری یک ارتباط مبتنی بر SIP

البته همان طور که قبلا هم ذکر کردیم در این فاز از پروژه ما به ارسال پیام های متنی بسنده کرده ایم (۲ـ۲). در واقع هدف بیش تر به سمت طراحی سیستم های توزیعی بوده است.

۲ـ۲ پیام رسانی

در SIP پیام ها بر دو نوع درخواست و پاسخ هستند. فرمان های به نام METHOD وجود دارند که نوع عملکرد پیام های SIP را تعیین می کنند. برای ارسال یک پیام متنی یا همان IM از متد MESSAGE استفاده می کنیم. دیاگرام کلی پیغام متنی در حالت ساده (یعنی حالتی که هر دو Client به یک Server متصل هستند) در شکل زیر آمده است.



شکل ۳-۲. نمونه دیاگرام پیغام متنی

۳ معماری شبکه توزیع ارسال و دریافت پیام

در این بخش نحوه ساخت شبکه توزیع پیشنهادی خود را به تفصیل توضیح خواهیم داد. ابتدا معماری عناصر شبکه توزیع را بیان کرده که این معماری در قالب نمودارهای متداول UML آمده است. پس از آن نیز توپولوژی شبکه توزیع، نحوه ارتباط گره ها با یکدیگر و در نهایت هم الگوریتم پیام رسانی در شبکه را شرح می دهیم.

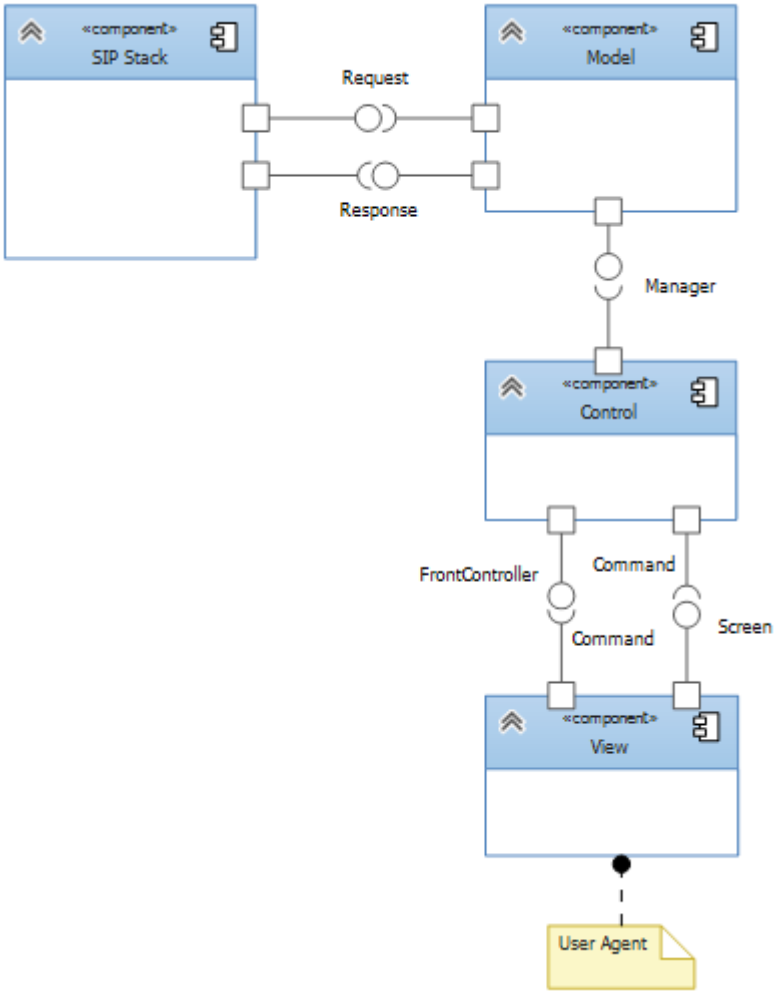
۳ـ۱ دیدگاه سطح مولفه سیستم

از دیدگاه کلی هدف از این پروژه ساخت یک شبکه توزیع شده برای ارسال و دریافت پیام متنی[۲] با استفاده از SIP است. ما با طراحی یک لایه برنامه کاربردی بر روی بستر پروتکل ارتباطی (SIP) این کار را انجام خواهیم داد. بنابراین در پایین ترین سطح ارتباطی، ما از یک چارچوب پیاده کننده SIP استفاده می کنیم و پیام ها را از طریق API که این چارچوب در اختیار ما قرار می دهد در شبکه توزیع منتقل می کنیم.

لایه مدل که بر روی چارچوب SIP قرار می گیرد در واقع منطق و کارکردهای اصلی برنامه ما را پیاده سازی می کند. با توجه به این که در امعماری کلی این برنامه ما از طرح MVC استفاده کرده ایم لایه مدل از طریق Controller با لایه نمایش که مسئول نمایش قسمت های مختلف است ارتباط برقرار می کند. دیدگاه سطح مولفه[۳] که در شکل ۱-۳ نشان داده شده است، معماری کلی سیستم را بازنمایی می کند. موجودیت های اصلی شبکه توزیع پیام همان طور که در بخش اول بیان شد، عبارتند از ۱- Client (UAC) و ۲ – Server (UAS). شکل ۱-۳ کلیت هر دو موجودیت سیستم می باشد که در بستر SIP واقع شده اند.

[۲] Instant Message (IM)

[۳] Component Diagram



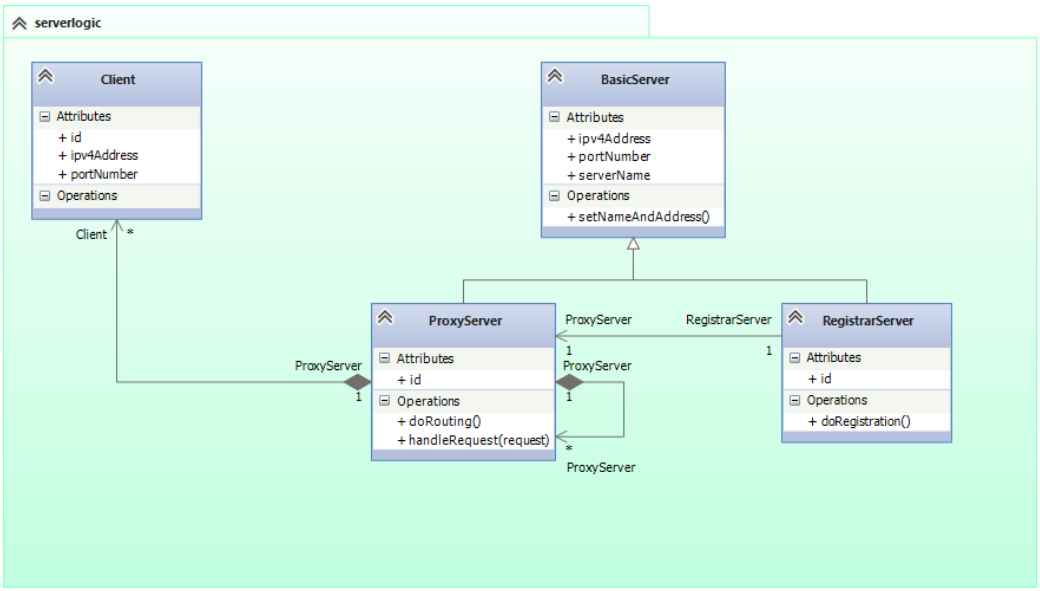
شکل ۱-۳ نمودار مولفه ها (Component Diagram) شبکه توزیع پیام

۲ـ۳ پیاده سازی موجودیت Server

برای پیاده سازی Serverها ما از یک معماری یک سان برای همه انواع موجود ا ستفاده کرده ایم. در حقیقت این تفکیک انواع در سطح منطقی و در قالب کلاس ها صورت گرفته است. این کار سبب می شود تا در صورت لزوم بتوانیم نقش یک Server را به راحتی عوض کنیم و همچنین هر Server بتواند اقدامات را برحسب نیاز سرویس دهی نماید.

برنامه سمت Server نوشته شده همه اقدامات مورد نیاز برای نقش های مختلف را شامل می شود. در ادامه نحوه پیاده سازی آن را شرح خواهیم داد. همان طور که در شکل ۴-۳

پیداســـت، Client ها به صـــورت گروهی با SIP Server های مختلف در ارتباط هســـتند، در نخســتین اتصـال، مشـخصـات آن ها از طریق Registrar Server دریافت شـده و پس از اعتبار سنجی[۴] در داخل لیست حاوی Client های Proxy Server مربوط ثبت می شوند. در شکل ۳-۲ نمودار کلاس لایه مدل Server در قالب UML آمده اســـت. توجه شـــود که جزئیات داخل کلاس ها (متدها و صفت ها) به طور کامل نشان داده نشده اند.

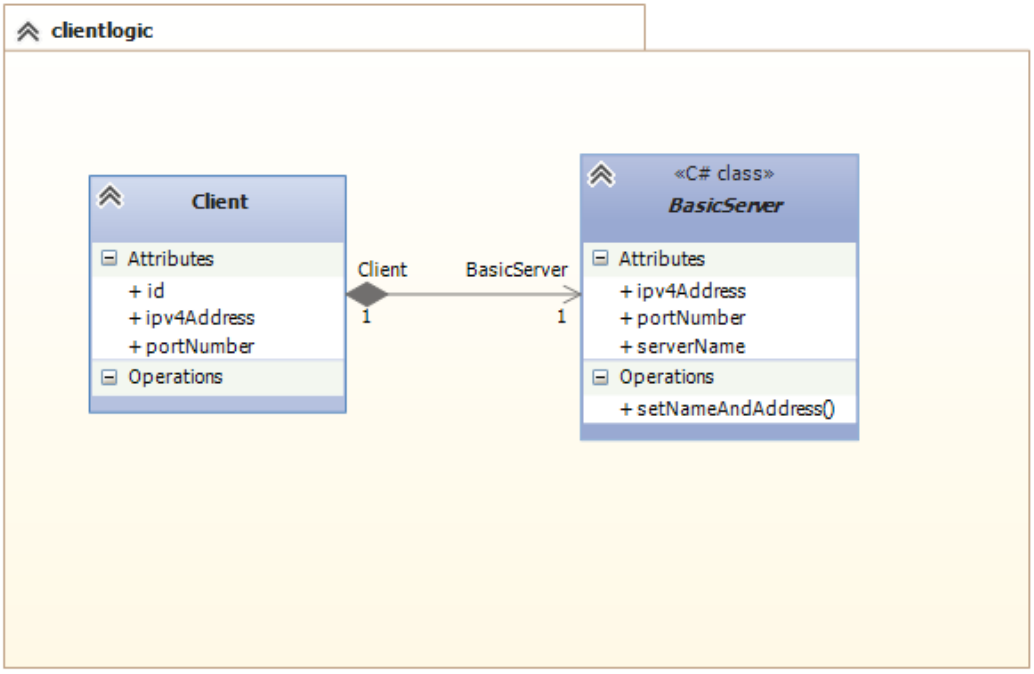


شکل ۲-۳. نمودار کلاس لایه مدل Server در قالب UML

۳ـ۳ پیاده سازی موجودیت Client

لایه مدل ســمت Client ســاده تر از قســمت Server اســت. در این لایه تنها اطلاعات Server که Client به آن مت صل می گردد نگهداری می شود و عملیات ا صلی نیز ار سال *پیام های متنی* و پیام های *ثبت نام* اســت. لایه نمایش نیز به صــورت یک واســط کاربری گرافیکی برای کاربر تعریف شده است. شکل ۳-۳ نمودار کلاس لایه مدل سمت Client را نشان می دهد.

[۴] Validation



شکل ۳-۳ نمودار کلاس لایه مدل سمت Client در قالب UML

۳ـ۴ بهم بست Server ها در شبکه توزیع (توپولوژی شبکه)

می توان از یک توپولوژی خاص برای چیدمان Server ها در شـــبکه اســـتفاده کرد. بنابراین سناریو افزودن Server به شبکه ی در حال استفاده، بایستی به نحوی تعیین شود که قواعد حاکم بر آن توپولوژی رعایت شود. مثلا چنان چه از توپولوژی گراف کامل ا ستفاده کنیم بایســتی مطمئن شــویم تمام گره ها (که در این جا منظور همان ســرویس دهنده ها اســت) به یکدیگر متصل شده اند و از وجود یکدیگر اطلاع دارند.

*** بروزر سانی شده در نگارش ۵:** در نسخه اولیه این طرح ما توپولوژی خا صی را برای شـــبکه توزیع در نظر نگرفتیم. هر چند این امر مزایایی به همراه داشـــت در نهایت برای بهبود الگوریتم ار سال و دریافت پیام (که در اینجا به آن مسیریابی اطلاق می شود)، از یک توپولوژی ترکیبی تحت عنوان ســـتاره – حلقه[۵] که در واقع ترکیبی از توپولوژی های ســـتاره و حلقه در

[۵] Ring - Star

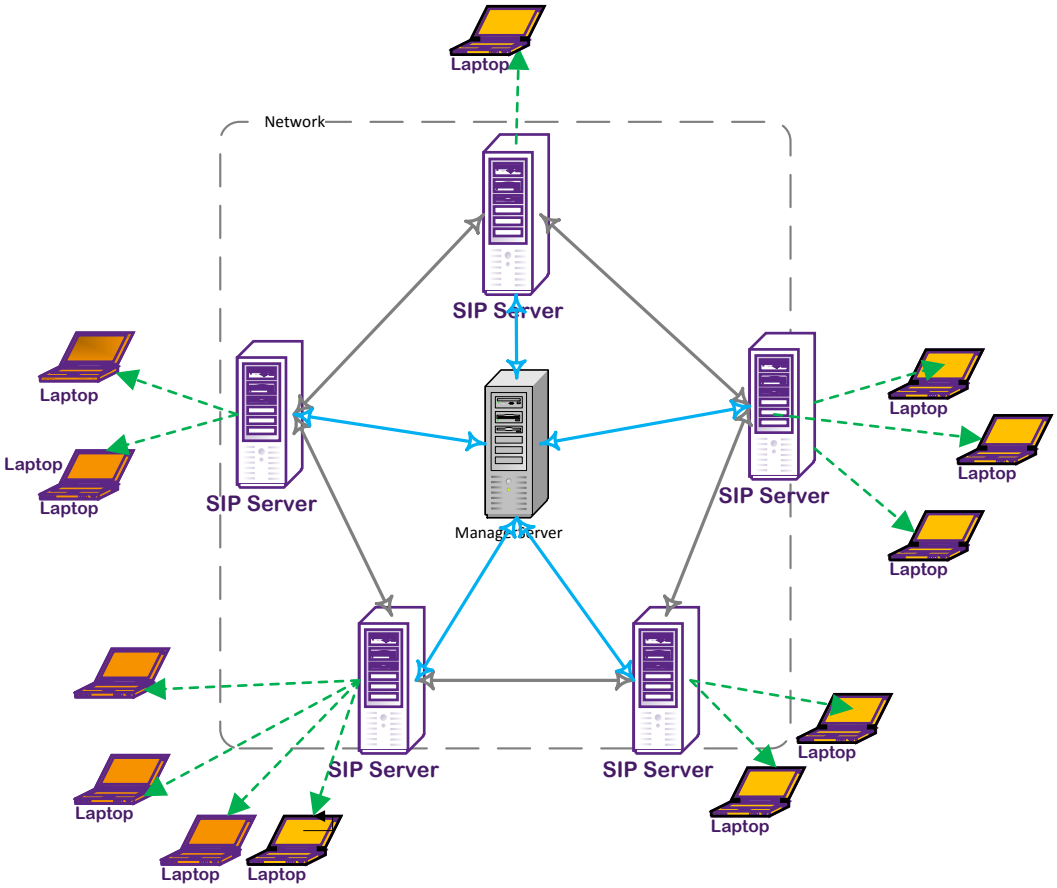
شبکه های کامپیوتری است، استفاده کرده ایم. به این ترتیب گره ها به نحوی به شبکه اضافه می شوند که توپولوژی ذکر شده رعایت شود.

به عبارت دیگر هنگامی که شبکه توزیع پیام در حال اجرا است می توانیم یک Server را اضافه به شبکه اضافه کرده و ارتباط آن را به صورت دستی با سرورهای مرکز، قبلی و بعدی موجود در شبکه توزیع برقرار سازیم. بنابراین حالت کلی توپولوژی شبکه به صورت یک گراف باقی خواهد ماند.

۳ـ۵ ارتباط Client ها با Server ها

به هر Server موجود در شبکه توزیع پیام تعدادی Client متصل هستند. در حال حاضر تعداد Client های متصل به هر Server را محدود نکرده ایم. در طرف مقابل هر Client در یک زمان به یک Server متصل خواهد بود. پس ارتباط Client-Server طرح ریزی شده ما، از نوع چند به یک[۶] است. شکل ۴-۳ ارتباطات بین Server ها با یکدیگر و نیز ارتباط بین Clientها با Server ها را مطابق توپولوژی در نظر گرفته شده و آن چه توضیح داده شد، به صورت فرضی به تصویر کشیده است.

[۶] Many to One



شکل ۴-۳. ارتباطات مختلف در شبکه توزیع با توپولوژی Ring-Star

۳ـ۶ سناریو و الگوریتم ها

در این قسمت سناریوهای لازم را استخراج کرده و الگوریتم کارا برای پیاده سازی هر سناریو، ارائه می دهیم. پس از تحلیل انجام شده، سه سناریو اصلی در ساختار پروژه شبکه توزیع تشخیص داده شد که عبارتند از **۱- سناریو ثبت نام Client**، ۲- **سناریو ارسال پیام و دریافت پاسخ مربوطه** و **۳ - سناریو افزودن Node به شبکه در حال اجرا**. در بخش های بعدی به تفصیل هر یک را بیان می کنیم.

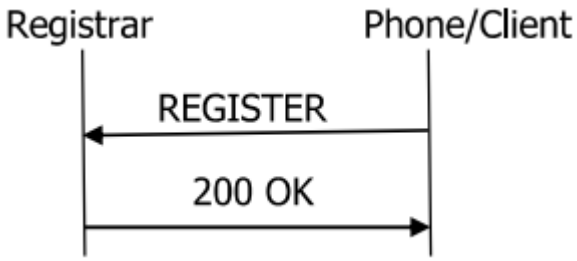
۳ـ۶ـ۱ سناریو ثبت نام Client

ثبت نام یعنی معرفی Client به Server و درج آدرس IP و شماره Port آن در Server.

در این جا گام های لازم برای ثبت نام یک Client به نام UAC A را توضیح می دهیم:

۱. UAC A به Registrar Server درخواســـت ثبت نام می دهد[۷] (می توان چند Server ثبت نام داشت).

۲. Registrar Server پس از بررسی درخواست و انجام اعتبار سنجی، UAC را به لیست Client های Server مربوط اضافه می کند و پیام 200 OK را به UAC A باز می فرستد.



شکل ۵-۳. سناریو ثبت نام

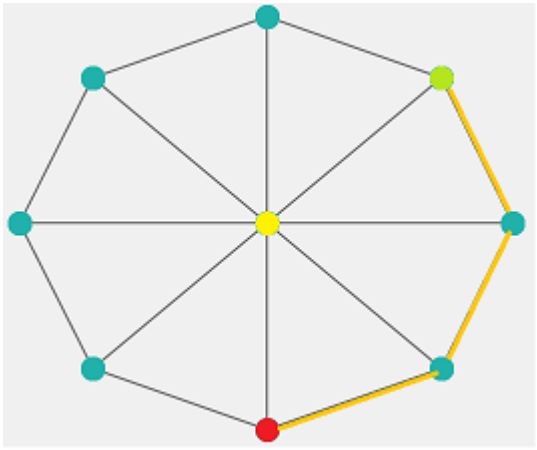
۳ـ۶ـ۲ سناریو مسیر یابی (ارسال و دریافت پیام)

در این قسمت به شرح روشی برای یافتن مسیر ارسال پیام در ارتباط بین دو Client می‌پردازیم. برای این منظور ابتدا Client ای که قصد دارد پیامی را ارسال کند، پیام خود را به Server ای که از قبل در آن ثبت نام کرده است ارسال می کند. Server، Client مقصد پیام را در لیست محلی Client های متصل به خود جستجو کرده چنان چه آن را در بین مشتری های محلی خود بیابد بلافاصله پیام را ارسال می کند و در صورتی که آن را نیابد از الگوریتم ساده زیر استفاده می کند.

❄ بروزرسانی شـــده در نگارش ۵: با توجه به این که گره ها در قالب توپولوژی Ring-Star بهم بسته شده اند. هر گره در شبکه توزیع، گره قبلی خود، گره بعدی خود و گره مرکزی را می شناسد. گره بعدی آخرین گره در شبکه همان اولین گره است. گره ای که پیام را دریافت می کند در صورتی که مقصد را در بین Clientهای متصل به خودش نیابد، آن را به گره بعدی Forward می کند. گره بعدی نیز همین عملیات را انجام می دهد. الگوریتم مادامی که Client مقصد یافت نشود و یا به گره ای که ارسال پیام از آن آغاز شده، نرسیده باشد، تکرار می گردد.

[۷] توجه شود که ارتباط بین UAC با Registrar Server در شکل نشان داده نشده است. این ارتباط در نخستین اتصال رخ می دهد و پس از آن ارتباط بین Client و یک Proxy Server برقرار می شود.

در حالت اول پیام به Client مقصد ارسال می شود و الگوریتم خاتمه می یابد. در حالت دوم با خطای Client Not Found الگوریتم پایان می یابد. شکل زیر مسیری را که الگوریتم در حالتی که Client مبدا در گره قرمز رنگ و Client مقصد در گره سبز رنگ است طی می کند، نشان می دهد. این مسیر با رنگ نارنجی مشخص شده است.



شکل ۶-۳ طرح واره ای از نحوه مسیر یابی

۳ـ۶ـ۳ تحلیل زمانی الگوریتم مسیر یابی

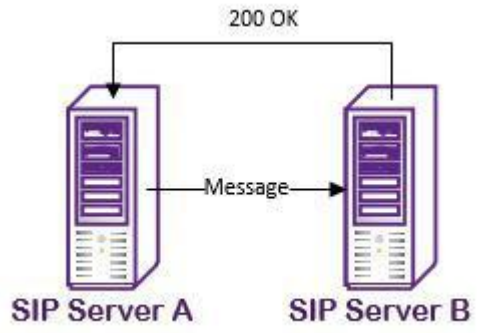
همان طور که پیداست این الگوریتم یه جست و جوی خطی ساده بوده و بنابراین از مرتبه ***O(n)*** می باشد که n همان تعداد گره ها یا ProxyServerها است. در صورتی که یکی از گره های روی حلقه از کار بیفتد مسیر یابی از طریق گره مرکزی ادامه می یابد. در صورتی که گره مرکزی از کار بیفتد اولین گره اضافه شده (گره با ID کوچک تر) جایگزین آن می شود. چگونگی این جایگزینی را توضیح می دهیم.

۳ـ۶ـ۴ سناریو افزودن Proxy Server

ابتدا طریقه اتصال دو Server با نام های فرضی A و B را بیان می کنیم. سپس نحوه اتصال به شبکه Ring-Star را بیان می کنیم. برای اتصال دو Server مطابق شکل ۳-۷ ابتدا Server A با در دست داشتن اطلاعات IP و PORT یکی از Serverهای شبکه به نام B و با توجه به صحت اطلاعات موجود و امکان ایجاد ارتباط با Server مذکور، برای آن درخواستی ارسال کرده و خود را به آن معرفی می کند.

Server B با دریافت پیام از طرف Server A برای آن یک پاسخ ارسال کرده و اطلاعات مورد نیاز برای ارتباط با Server A را در لیست Serverهای متصل به خود ذخیره می کند.

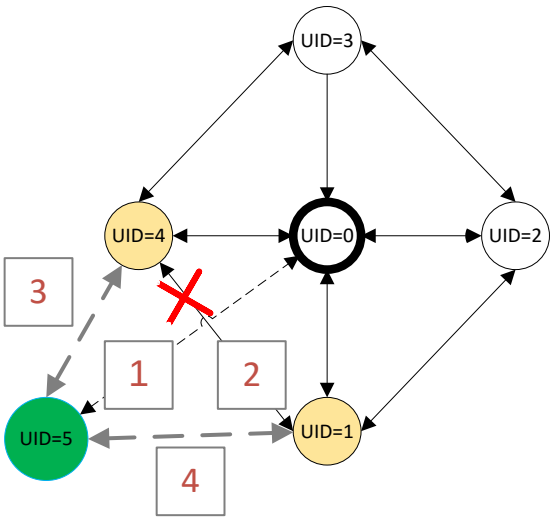
در پایان Server A پس از دریافت پاسخ OK از طرف B و اطمینان از صحت برقراری ارتباط، اطلاعات مورد نیاز برای ارتباط با B را در لیست Serverهای متصل به خود ذخیره می کند.



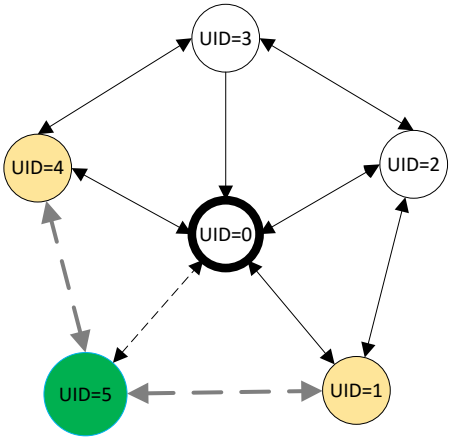
شکل ۷-۳. طرح واره ای از نحوه اتصال دو سرویس دهنده

❄ بروزرسانی شده در نگارش ۵: اتصال به شبکه توزیع Ring-Star: مطابق توپولوژی توضیح داده شده، هر گره در این شبکه با سه گره دیگر در ارتباط است: ۱- گره قبلی (Pervious or Predecessor)، ۲- گره بعدی (Next or Successor) و ۳- گره مرکزی (Center or Manager)[۸]. در شروع فرایند اتصال به شبکه، Serverای که قرار است به شبکه متصل شود ارتباط خود را با Serverای که در نقش Manager (Server مرکزی) قرار دارد مطابق سناریو شکل ۷-۳ برقرار می سازد. در مرحله بعد Server مرکزی بروز رسانی اتصال های Serverهای قبلی و بعدی و جدید را برای هر سه Server انجام می دهد ترتیب اجرای این مراحل با شماره های ۱ تا ۴ در شکل ۸-۳ آمده است. باید توجه شود که Server جدید همواره به انتهای زنجیره متصل می شود؛ یعنی بین آخرین Server موجود در شبکه و اولین Server. شکل ۹-۳ نیز وضعیت شبکه را پی از اجرای مراحل ۴ گانه شکل قبلی نشان می دهد.

[۸] باید به این نکته توجه نمود که منظور از این اتصالات، اتصال منطقی گره ها به یکدیگر است. ممکن بین هر دو گره منطقی چندین و چند گره فیزیکی وجود داشته باشد. در واقع ما یک Overlay در لایه کاربرد پیاده سازی می کنیم. کلیه المان های شبکه توزیع توصیف شده در سطح منطقی هستند.



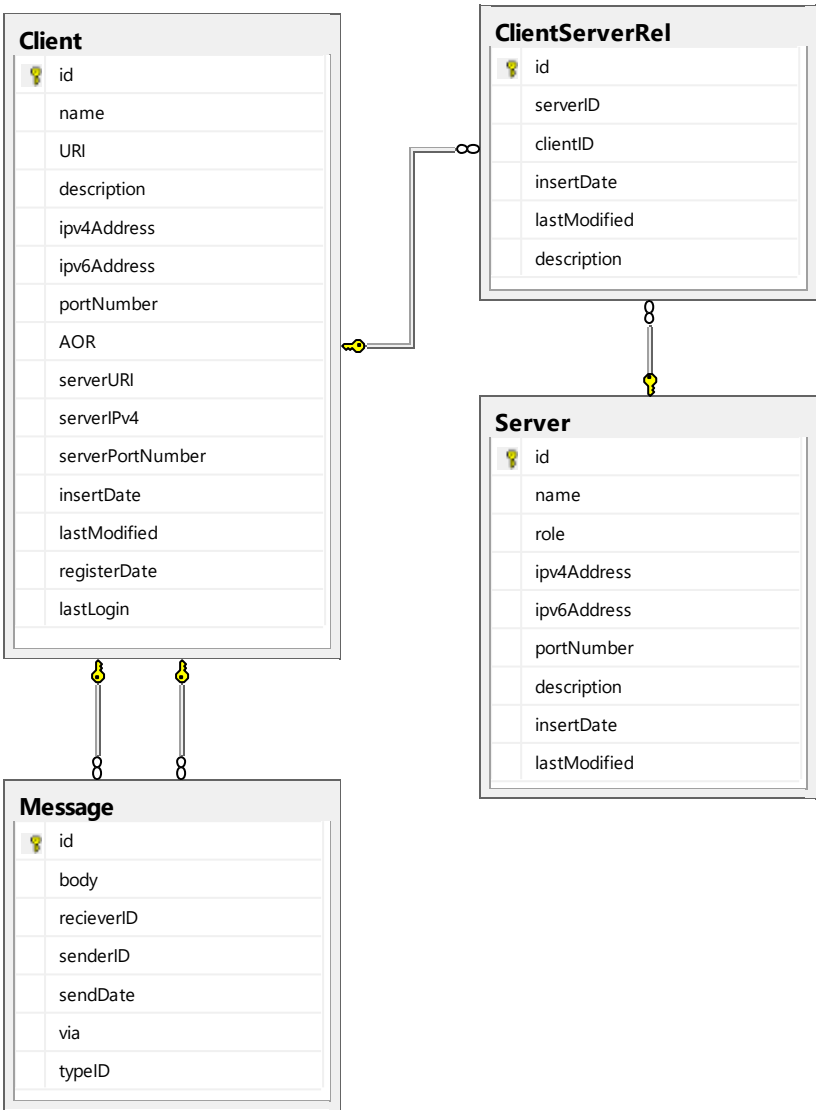
شکل ۸-۳. مراحل اضافه شدن گره جدید به انتهای شبکه Ring-Star



شکل ۹-۳ گره اضافه شده به شبکه توزیع

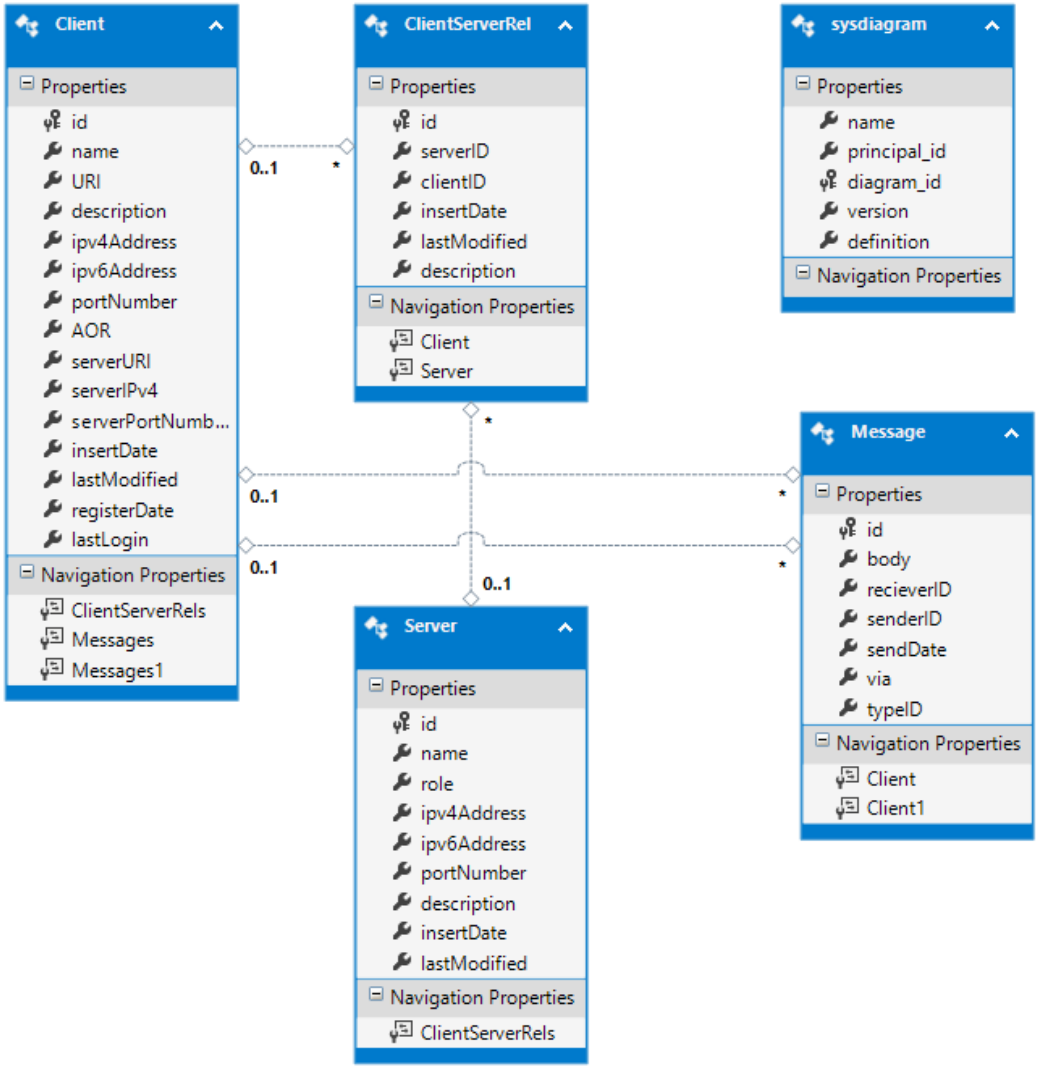
۳ـ۷ مدل بانک اطلاعات برنامه

برای ذخیره مشخصات client ها و Server ها و ارتباط آن ها و نیز پیام های ارسالی (در صورت لزوم)، نیاز به استفاده از یک ساختار ذخیره سازی است. در ایجا ما با پیروی از مدل رابطه ای از یک پایگاه داده ساده SQL برای نیل به اهداف ذکر شده بهره جسته ایم. شمای این پایگاه داده در شکل زیر آمده است. این پایگاه داده در سمت Server نگهداری می شود و از نرم افزار SQL Server 2014 شرکت مایکروسافت در پیاده سازی و اجرای آن استفاده شده است.



شکل ۱۰-۳. شمای پایگاه داده شبکه توزیع

ا سامی جداول و فیلدهای ا ستفاده شده وا ضح بوده و نیاز به تو ضیح ندارد. همچنین ممکن ا ست در طی فاز های بعدی بخش هایی ا ضافه، حذف و یا بروزر سانی شود. شمای فوق سپس با ا سفاده از ORM به کلاس های برنامه C# نگاشت شده است که در شکل زیر می توان مدل کلاس های به دســـت آمده از نگاشـــت را به همراه ارتباطات بین آن ها دید. بدین ترتیب قابلیت استفاده از این کلاس ها در برنامه فراهم شده است.



شکل ۱۱-۳. مدل داده شبکه توزیع در قالب نمودار ER

۴ بررسی معیارها و اهداف سیستم های توزیعی

در این بخش ق صد داریم طرح و مدل ارائه شده را مورد برر سی قرار داده و میزان براورده کردن اهداف یک سیستم توزیع شده را بسنجیم. برای این منظور ابتدا لازم است جایگاه معماری سیستم پیشنهادی را در معماری های شناخته شده برای سیستم های توزیع شده،

بیان کنیم. سپس اهداف سیستم های توزیعی را شرح می دهیم. این اهداف، قابلیت دستیابی[۹]، شفافیت[۱۰] ، باز[۱۱] و مقیاس پذیر[۱۲] بودن سیستم هستند.

۱ـ۴ بررسی جایگاه معماری

سه معماری متداول ارایه شده برای سیستم های توزیعی عبارتند از ۱- Centralized، ۲- Decentralized و ۳- Distributed. با توجه به این که کلیه گره هایی داخل شبکه توزیع از ظرفیت پردازشی و امکانات یکسانی برخوردار هستند و از این جهت برتری نسبت به یکدیگر ندارند و نیز با توجه به نحوه اتصال به یکدیگر که از توپولوژی خاصی استفاده می کنند، می توان گفت شبکه ای از گره های همگن را در اختیار داریم.

عملکرد[۱۳] توزیع شده در این شبکه توزیع را می توان *ارسال پیام* در نظر گرفت که در رابطه با آن همه گره ها نقش یکسانی را ایفا می کنند. همچنین گره ای که به عنوان گره مرکزی در سیستم وجود دارد، همان قابلیت های دیگر گره ها را داراست با این تفاوت که در برخی امور اولویت آن بالاتر است و به عنوان هماهنگ کننده و تسهیل کننده وارد عمل می شود. به طور خاص وظایف گره مرکزی عبارتند از:

۱- تولید شناسه (ID) عددی یکتا برای گره ها در هنگام اتصال،
۲- بروز رسانی اتصال ها در هنگام افزودن گره،
۳- نگهداری لیست کلیه گره های فعال در شبکه،
۴- فراهم سازی مکانیسم Cashing در مسیر یابی،
۵- ایجاد ارتباط بین Partition های شبکه هنگامی که یکی از گره ها به دلیل خرابی از دسترس خارج می شود (افزایش تحمل پذیری خطا).

[۹] Accessibility
[۱۰] Transparency
[۱۱] Openness
[۱۲] Scalability
[۱۳] Function

با توجه به این که قابلیت های فوق به در همه گره ها تعبیه شده است، در صورت از کار افتادن گره مرکزی، به صورت پیش فرض گره با کوچکترین شناسه، نقش گره مرکزی را بر عهده می گیرد.

در نهایت می توان نتیجه گرفت معماری ارایه شده از نوع **Decentralized** ساختار یافته (Structured P2P) است.

۴ـ۲ قابلیت دستیابی

سیستم توزیعی پیشنهادی ما در این جا قابلیت دسترسی دارد. بدین معنا که در هر مکان Client می تواند با انجام عملیات ثبت نام از منابع Server ها برای ارسال پیام استفاده کند. همچنین می توان Server های را به شبکه توزیع افزود.

۴ـ۳ شفافیت

بر مبنای تعریف شفافیت، سیستم های توزیعی می بایست از دید کاربران تا حد امکان شفاف، ساده و فاقد دشواری های پیاده سازی باشند. در سیستم مورد مطالعه ی ما، کاربران را می توان افرادی در نظر گرفت که با در اختیار داشتن یک نرم افزار پیام رسان ساده، اقدام به برقراری ارتباط و ارسال پیام به یکدیگر را دارند. شفافیت یک سیستم جنبه های مختلفی را شامل می شود:

۴ـ۳ـ۱ شفافیت دستیابی

زمانی که یک کاربر تصمیم به ارسال پیام به کاربر دیگری می کند، تنها کاری که انجام داده (پس از ثبت نام در سیستم) تعیین مخاطب مورد نظر، نوشتن متن پیام و ارسال آن است. پس می توان گفت شیوه دستیابی به سرویس دهنده ها و سایر کاربران از دید او پنهان خواهد بود.

۴ـ۳ـ۲ شفافیت مکانی

کاربران در این طرح هیچ گونه اطلاعی از مکان فیزیکی ســـرویس دهنده ها و ســـایر کاربران ندارند و صرفاً با استفاده از نام و یا آدرس (IP) (چون در حال حاضر DNS در نظر گرفته نشده است)، با آن ها ارتباط برقرار می کنند.

۴ـ۳ـ۳ شفافیت مهاجرتی

در این بخش باید به این سوال پاسخ داد که آیا هنگامی که مکان فیزیکی یک سرویس دهنده تغییر می کند، این تغییر از دید کاربر متصل به آن سرور پنهان می ماند یا خیر. در طرح ارائه شده، هر کاربر پس از ثبت نام در یک سرویس دهنده، تنها به همان سرویس دهنده متصل بوده و پیام های خود را مبادله می کند. در واقع تنها آدرســی که با آن ســروکار خواهد داشــت، آدرس IP سرویس دهنده خودش ا ست. در صورتی که یک سرویس دهنده متوجه تغییر مکان فیزیکی خود گردد (IP آن تغییر کند)، آدرس IP جدید خود را به نرم افزار پیام رســـان کاربران متصــل به آن ارســـال می کند (بدون اطلاع شــخص کاربر). در ادامه نرم افزارهای پیام رســـان، آدرس سرویس دهنده خود را بروزرسانی و ذخیره می کنند. از این به بعد نرم افزار پیام رسان با آدرس IP جدید اتصال خود را انجام خواهد داد.

۴ـ۳ـ۴ شفافیت جابجایی

در مورد این که تغییر مکان ســـرویس دهنده ای که در حال اســـتفاده اســـت، چه تاثیری در سیستم توزیعی خواهد دا شت، در حال حا ضر فرض را بر ثابت بودن مکان سرویس دهنده ای که تعریف شده است، گذاشته ایم. لذا می توان گفت شفافیت جابجایی برای این سیستم دیده نشده است.

۴ـ۳ـ۵ شفافیت در تکرار

در طرح ارائه شــده، نگرانی چندانی درباره این موضــوع نخواهیم داشــت. زیرا داده های موجود در سیستم محدود به آدرس های سرویس دهنده ها و کاربران متصل به هر کدام از آن ها ا ست و همانطور که گفته شد، هر کاربر فقط با یک سرویس دهنده در ارتباط ا ست. با این حال در صورت وجود تکرار، این مسئله به سادگی از دید کاربران مخفی خواهد ماند.

۴ـ۳ـ۶ شفافیت هم روندی

هنگامی که کاربری اقدام به ارسال پیام به کاربر دیگر می کند، از سوی سیستم پیغامی مبنی بر موفقیت و یا عدم موفقیت در ار سال پیام به او داده خواهد شد. حال این که چه تعداد کاربر هم اکنون در سیستم حضور داشته و درحال استفاده از سیستم هستند، همگی از دید او پنهان خواهد بود.

۴ـ۳ـ۷ شفافیت خرابی

این شفافیت تا حدودی برای دیده شده است. برای مسئله خرابی گره مرکزی با مشکل مواجه نمی شویم. اما در صورتی که گره ای از د سترس شبکه خارج شود، ارتباط Client ها متصل به آن با شبکه توزیع انتقال پیام قطع خواهد شد. برای این مسئله می توان یک سرویس دهنده پیش فرض جایگزین در نظر گرفت و یا ثبت نام را همزمان در چندین محل انجام داد (ارتباط چند به چند بین مشتری ها و سرویس دهنده ها در شبکه توزیع)، که در حال حاضر به دلیل محدودیت های زمانی پیاده سازی، این راه حل ها را در طرح پیشنهادی مطرح ننموده ایم.

۴ـ۴ باز بودن

در صورتی که کاربری با استفاده از پروتکل ارتباطی SIP (نه الزاماً نرم افزار توسعه داده شده ی ما) قصد اتصال به این سیستم توزیع شده را داشته باشد، این امکان وجود دارد (در سمت مشتری باز بودن وجود دارد)، اما تنها Serverهایی که طراحی ما را پیاده سازی کرده باشند قادر به اتصال به شبکه توزیع هستند (عدم باز بودن در سمت سرور).

۴ـ۵ مقیاس پذیری

در استاندارد SIP فیلدی تحت عنوان MaxFroward در سرایند پیام های ارسالی وجود دارد که توصیه شده است عدد ۷۰ به صورت پیش فرض در آن قرار داده شود. هرگاه پیام از UA به UA دیگری Forward شود از این فیلد یک واحد کم می شود. در صورتی که مقدار این فیلد به صفر برسد و مقصد یافت نشود، آن گاه پیام در شبکه گم می شود. مطابق استاندارد بیشینه عددی که می توان در این فیلد قرار داد عدد ۲۵۵ است (۱ بایت برای این فیلد در نظر

گرفته شـده اسـت)؛ یعنی مجاز هسـتیم حداکثر ۲۵۵ بار یک پیام را Forward کنیم. الگوریتم مسیریابی ما از مرتبه O(n) است بنابراین در بدترین حالت بعد از ۲۵۵ بار ارسال یک پیام، آن پیام از بین می رود. در نتیجه حداکثر تعداد گره ای که می توان در Ring قرار داد به نحوی که مطمئن باشیم پیام از بین نمی رود ۲۵۵ عدد است.

تا این مرحله مقیاس پذیری ما به تعداد ۲۵۵ گره در شبکه توزیع محدود شده است. با فرض این که به هر گره حداکثر ۱۰,۰۰۰ عدد Client بتوان متصـل کرد. شـبکه حدودا از ۲,۵۵۰,۰۰۰ مشـتری پشـتیبانی می کند که عدد بالایی اسـت. در نسـخه های آتی طرح راهکارهایی برای افزایش ظرفیت شبکه و مقیاس پذیری آن پیشنهاد می گردد.

۵ مشخصات فنی پروژه

این پروژه به زبان برنامه نویسی C# و در محیط توسعه مجتمع Microsoft Visual Studio 2015 تولید شده است. جهت اجرای صحیح پروژه و نیز مشاهده و تغییر کد برنامه نرم افزارها و ابزار های زیر ضروری است.

- Microsoft .NET Framework 4.5.2 or Upper
- کتابخانه LumiSoft.Net
- Microsoft Visual Studio 2015 (for Rebuild Purposes)

البته برای اجرا و استفاده از پروژه به عنوان کاربر کافی است بر روی سیستم عامل ویندوز خود ابزار .NET Framework 4.5.2 و یا بالاتر را نصب داشته باشید و نیازی به سایر موارد نیست.

۶ منابع و ماخذ

[1] A. S. Tanenbaum and M. V. Steen, DISTRIBUTED SYSTEMS principles and paradigms, Pearson Prentice Hall, 2007.

[2] N. W. Group, " SIP: Session Initiation Protocol," 2002. [Online]. Available: https://www.ietf.org/rfc/rfc3261.txt .