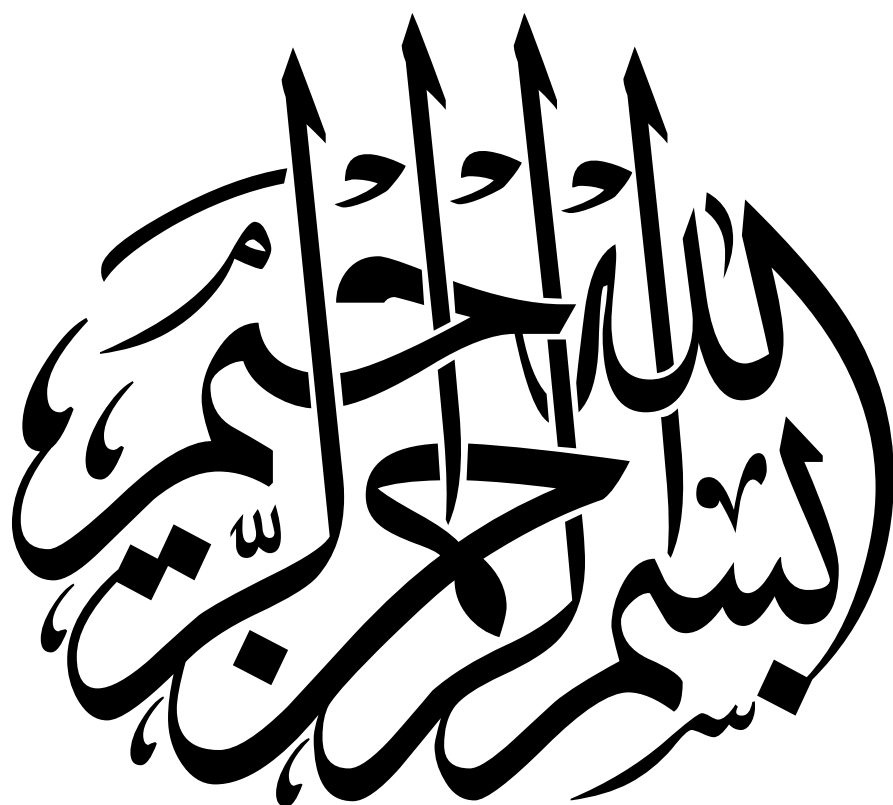






دانشگاه صنعتی شریف

دانشکده مهندسی برق

پایان نامه دکتری

گرایش الکترونیک

عنوان:

ساختارهای مسیریابی برای محیط های مجازی گسترده با تعداد کاربران زیاد

نگارش:

بهمنوش حریری

اساتید راهنما :

دکتر محمد حسین علوی

دکتر محمد رضا پاکروان

مرداد ۱۳۸۸

به نام خدا
دانشگاه صنعتی شریف
دانشکده مهندسی برق
رساله دکتری

عنوان: ساختارهای مسیریابی برای محیط‌های مجازی گسترده با تعداد کاربران زیاد

نگارش: بهنوش حریری

کمیته ممتحنین:

امضاء..... استاد راهنما: دکتر محمد حسین علوی

امضاء..... استاد راهنمای همکار: دکتر محمدرضا پاکروان

امضاء..... استاد ممتحن داخلی: دکتر بیژن وثوقی وحدت

امضاء..... استاد ممتحن داخلی: دکتر بابک حسین خلیج

امضاء..... استاد ممتحن خارجی: دکتر جمال گلستانی

امضاء..... استاد ممتحن خارجی: دکتر شروین شیرمحمدی

امضاء..... استاد ممتحن خارجی: دکتر سید مصطفی صفوی

تاریخ:.....

تقدیر و تشکر

دل گرچه در این بادیه بسیار شتافت یک موی ندانست ولی موی شکافت

اندر دل من هزارخورشید بتافت آخر به کمال ذره‌ای راه نیافت

"منت خدای را عزوجل که طاعتش موجب قربت است و به شکر اندرش مزید نعمت" اکنون که در پرتو الطاف پروردگار مهربان، فراز و نشیب های متعدد پشت سر نهاده شده تا انجام این رساله به اتمام برسد، لازم می دانم تا از تمامی کسانی که در گذر از این مسیر دشوار مرا یاری و همراهی نموده اند قدردانی نمایم که بدون شک عبور از این گذرگاه بدون تکیه بر حمایت های بی دریغ آنها برای من ناممکن و دور از دسترس بوده است. جناب آقای دکتر علوی برای من الگوی استادی فداکار و متعهد هستند که آنچه من در این سال ها از ایشان آموخته ام ، سرلوحه آینده ام خواهد بود. استاد گرامی، آقای دکتر پاکروان درس های بسیاری را در بهترین زیستن به من آموخته اند و در نقاطی که ادامه راه برای من دشوار نموده است دست یاریگرشان مسیر را برای من هموار نموده و به من توان و امید ادامه دادن بخشیده است. انجام این رساله بدون وجود کمک های بی دریغ و همه جانبه جناب آقای دکتر شیرمحمدی میسر نبوده است که در اینجا خالصانه ترین مراتب سپاسگزاری خود را به ایشان تقدیم می دارم. از اساتید ممتحن نیز که برای حضور در جلسه دفاعیه من قبول زحمت نموده اند بسیار قدردانی می نمایم. در نهایت، وجود تمام دوستانی را که در این سال ها در کنار من بوده اند و وجودشان به من شور زندگی بخشیده است، ارج می نهم.

چکیده :

محیط های مجازی گسترده امکان تعامل بلادرنگ کاربران بسیار زیادی را که از طریق شبکه اینترنت به هم متصل هستند، فراهم می آورند. برای ایجاد احساس واحد از زمان و مکان برای کاربران محیط مجازی، تمام کاربران باید با تأخیری اندک از تغییرات محیط مجازی و نتایج تعاملات سایر کاربران مطلع شوند. این اطلاع رسانی با توزیع پیام های به روز رسانی میان کاربران محیط مجازی انجام می شود. نکته قابل توجه آن است که در حضور تعداد بسیار زیاد کاربران مبادله حجم بسیار زیاد پیام های به روز رسانی، یکی از چالش های بزرگ در طراحی محیط های مجازی است. استفاده از ساختارهای غیر متمرکز راه حلی برای گریز از این مسأله است. ولی این ساختارها نیز به نوبه خود به دلیل عدم وجود کنترل مرکزی مسایلی را ایجاد می کنند که نیاز به بررسی دارند. هدف از این رساله ارایه ساختارهای گسترده برای محیط های همکاری مجازی با تعداد کاربران بسیار زیاد است. پشتیبانی قابلیت توسعه و کمینه سازی تأخیر از اهداف مهم در این طراحی در نظر گرفته شده اند. شبیه سازی ها نشان دهنده کارایی این ساختارها در انتقال پیام های به روز رسانی در محیط های مجازی گسترده با میزان تأخیر نسبتاً کم در محدوده مورد قبول، می باشد.

کلمات کلیدی- محیط های مجازی با کاربران بسیار زیاد، شبکه های نظیر به نظیر، الگوریتم های گسترده، بهینه سازی گروه مورچه ها، منحنی هیلبرت، درخت KD، مسیر یابی هندسی در شبکه های سیمی

فهرست مطالب

۱.مقدمه	۲
۱.۱. انگیزه های تحقیق	۸
۲.۱. چالش ها	۹
۳.۱. نقاط اصلی نوآوری	۱۱
۴.۱. ساختار کلی رساله	۱۲
۲.مروری بر کلیات و تحقیقات مرتبط	۱۴
۱.۲. ساختارهای شبکه در محیط های مجازی	۱۵
۱.۱.۲. ساختار مشترک- سرور	۱۵
۲.۱.۲. ساختار نظیر به نظیر	۱۵
۳.۱.۲. ساختار ترکیبی	۱۹
۲.۲. مدیریت ناحیه مورد علاقه	۲۰
۱.۲.۲. مدل فضایی	۲۱
۲.۲.۲. مدل های چند پخشی در لایه کاربرد	۲۳
۳.۲.۲. مدل های ترکیبی	۲۴
۴.۲.۲. مدل مورد استفاده	۲۵
۳.تعریف و مدل سازی مسأله	۲۸
۱.۳. پهنای باند و کیفیت پیوندهای شبکه زیرساخت	۲۹
۲.۳. ماتریس ترافیک	۳۰

۳۰.....	۱.۲.۳. توزیع کاربران
۳۲.....	۲.۲.۳. ارتباط تقاضای تبادل پیام های به روز رسانی با فاصله کاربران
۳۳.....	۳.۳. ماتریس تأخیر
۳۵.....	۴.۳. تابع هدف
۳۹.....	۵.۳. یافتن پاسخ بهینه
۴۱.....	۱.۵.۳. تعاریف
۴۳.....	۲.۵.۳. مدل سازی مسأله
۴۹.....	۴. ساختار مبتنی بر DHT
۴۹.....	۱.۴. مقدمه ای بر DHT
۵۱.....	۲.۴. استفاده از DHT در محیط های مجازی
۵۴.....	۳.۴. روش های کاهش تأخیر مسیریابی در DHT
۵۵.....	۱.۳.۴. نسبت دادن ID های مناسب
۶۲.....	۲.۳.۴. انتخاب مسیرها با توجه به تأخیر
۶۴.....	۳.۳.۴. جستجوی مسیرهای مناسب با روش گروه مورچه ها
۶۹.....	۴.۴. نتایج شبیه سازی
۷۵.....	۵. ساختار مبتنی بر منحنی هیلبرت
۷۹.....	۱.۵. اندیس دهی چند بعدی با منحنی هیلبرت
۸۳.....	۲.۵. مسیریابی به کمک تبدیل هیلبرت
۸۶.....	۳.۵. الگوریتم مسیریابی

۴.۵. نتایج شبیه سازی	۸۹
۱.۴.۵. قابلیت توسعه	۹۱
۶. ساختار مبتنی بر درخت KD	۹۴
۱.۶. منحنی پر کننده فضای ترتیب Z	۹۶
۲.۶. درخت های جستجوی چند بعدی دودویی (درخت KD)	۹۸
۳.۶. نمایش منحنی Z در قالب درخت KD	۱۰۳
۴.۶. الگوریتم مسیر یابی	۱۰۴
۵.۶. نتایج شبیه سازی	۱۰۸
۱.۵.۶. قابلیت توسعه	۱۰۹
۷. ساختار مبتنی بر مسیریابی هندسی	۱۱۲
۱.۷. اصول LSH	۱۱۸
۲.۷. نگاشت سلسله مراتبی	۱۲۱
۳.۷. نگاشت با چند جدول	۱۲۲
۴.۷. الگوریتم مسیریابی	۱۲۳
۵.۷. مدیریت تغییرات در شبکه	۱۲۸
۱.۵.۷. پیوستن نقطه جدید به شبکه	۱۲۹
۲.۵.۷. حرکت نقاط	۱۲۹
۳.۵.۷. خارج شدن یک نقطه از شبکه	۱۳۳
۶.۷. نتایج شبیه سازی	۱۳۳

۱.۶.۷. کارایی مسیر یابی ۱۳۴

۲.۶.۷. قابلیت توسعه ۱۳۵

۳.۶.۷. حرکت نقاط ۱۳۶

۸. نتیجه گیری و پیشنهادات ۱۴۰

۹. مراجع ۱۴۸

فهرست جدول ها

جدول ۵-۱ ارتباط میان مختصات نقاط و کلید آنها در هر حالت ۸۱

جدول ۸-۱ مقایسه کلی و رده بندی ساختارهای پیشنهادی ۱۴۴

فهرست شکل ها

- شکل ۱-۱ ساختار تبادل پیام ها در محیط مجازی ۶
- شکل ۱-۳ توزیع کاربران در محیط مجازی الف) HPA ب) HPN ج) CCP ۳۱
- شکل ۲-۳ توزیع کاربران در محیط مجازی الف) سه بعد ب) صفحه xy ج) صفحه yz د) صفحه xz ۳۳
- شکل ۳-۳ دیاگرام شبکه جریان چند مرسوله ای ۴۵
- شکل ۱-۴ نمونه مسیریابی یک بسته در Pastry ۵۱
- شکل ۲-۴ تخصیص شناسه (الف به شکل تصادفی ب) مبتنی بر مکان ۵۶
- شکل ۳-۴ منحنی هیلبرت مرتبه ۱ تا ۳ ۵۷
- شکل ۴-۴ منحنی هیلبرت سه بعدی مرتبه ۱-۳ ۵۸
- شکل ۵-۴ نگاشت مکان کاربران به وسیله منحنی هیلبرت ۵۹
- شکل ۶-۴ فاصله نرمالیزه در فضای محیط مجازی و فضای شناسه ها ۶۰
- شکل ۷-۴ توزیع انحراف فاصله ۶۱
- شکل ۸-۴ جدول مسیریابی ساختار DHT ۶۸
- شکل ۹-۴ میانگین در حرکت تأخیر (ms) برای شبکه Pastry، روش پیشنهادی و ساختار بهینه ۷۰
- شکل ۱۰-۴ تابع توزیع تأخیر (ms) برای شبکه Pastry، روش پیشنهادی و ساختار بهینه ۷۱
- شکل ۱۱-۴ تأخیر مسیرهای شبکه در مقابل تقاضای ترافیک در آنها در ساختار پیشنهادی ۷۳
- شکل ۱۲-۴ تأخیر مسیرهای شبکه در مقابل تقاضای ترافیک در آنها در شبکه Pastry ۷۳
- شکل ۱-۵ ربع فضاهای تشکیل شده توسط منحنی هیلبرت ۷۹
- شکل ۲-۵ نمایش درختی منحنی سه بعدی هیلبرت ۸۲
- شکل ۳-۵ شبه کد نگاشت مرحله ای نقاط روی منحنی هیلبرت ۸۳

- شکل ۴-۵ نمونه ای از تعریف همسایگی ها برای یک نقطه درخت دو بعدی هیلبرت ۸۵
- شکل ۵-۵ جدول مسیریابی ساختار هیلبرت ۸۶
- شکل ۶-۵ شبه کد الگوریتم مسیریابی بر مبنای منحنی هیلبرت ۸۸
- شکل ۷-۵ هیستوگرام گام های ارسال بسته ها ۹۰
- شکل ۸-۵ متوسط تعداد گام های مسیر بسته ها بر حسب تعداد نقاط شبکه ۹۱
- شکل ۱-۶ نگاشت Z ۹۷
- شکل ۲-۶ منحنی پر کننده فضای Z ۹۷
- شکل ۳-۶ تقسیم بندی فضا در دو بعد برای توزیع غیر یکنواخت نقاط ۱۰۰
- شکل ۴-۶ درخت KD متناظر ۱۰۰
- شکل ۵-۶ شبه کد تابع تشکیل درخت KD ۱۰۱
- شکل ۶-۶ مثالی از درخت KD متناظر با منحنی Z ۱۰۴
- شکل ۷-۶ جدول مسیریابی ساختار درخت KD ۱۰۶
- شکل ۸-۶ شبه کد الگوریتم مسیریابی بر مبنای درخت KD ۱۰۷
- شکل ۹-۶ هیستوگرام گام های ارسال بسته ها ۱۰۸
- شکل ۱۰-۶ متوسط تعداد گام های مسیر بسته ها بر حسب تعداد نقاط شبکه ۱۰۹
- شکل ۱-۷ فرآیند سلسله مراتبی نگاشت در دو مرتبه ۱۲۲
- شکل ۲-۷ شبه کد الگوریتم مسیریابی هندسی ۱۲۷
- شکل ۳-۷ تغییر در همسایگی ها هنگام حرکت یک نقطه ۱۳۲
- شکل ۴-۷ هیستوگرام گام های ارسال بسته ها ۱۳۴
- شکل ۵-۷ متوسط تعداد گام های مسیر بسته ها بر حسب تعداد نقاط شبکه ۱۳۶

شکل ۶-۷ سازگاری توپولوژی برای یک نقطه تصادفی ۱۳۷

شکل ۷-۷ توزیع سازگاری توپولوژی در تمام شبکه ۱۳۸

فصل اول

مقدمه

انگیزه های تحقیق

چالش ها

نقاط اصلی نو آوری

ساختار کلی رساله

واقعیت مجازی^۱ که در سال های اخیر مورد توجه بسیاری از تحقیقات بوده است در اصطلاح به محیطی گفته می شود که برای کاربران امکان قدم نهادن در یک دنیای مجازی و تعامل در آن را فراهم آورد. دنیای مجازی نزدیک به واقع باید بتواند تمام آنچه را در دنیای واقعی رخ می دهد، به نحوی شبیه سازی نماید. کاربرانی که در یک محیط مجازی قرار می گیرند باید بتوانند واکنش های محیط را در هنگام تعاملات خود به صورت بلادرنگ احساس نمایند. با رشد شبکه های انتقال داده و افزایش توانایی های کامپیوترهای شخصی، روند تمایل به استفاده از سیستم های گسترده شبیه سازی کامپیوتری که به تعدادی کاربر اجازه تعامل در یک محیط مجازی را می دهند، افزایش یافته است. در چنین ساختاری، محیط شبیه سازی شده در چندین کامپیوتر اجرا می شود که این کامپیوترها از طریق یک شبکه کامپیوتری مانند اینترنت به هم متصل هستند. بدین صورت کاربرانی که از نظر جغرافیایی در مکان های مختلف پراکنده هستند، می توانند به صورت بلادرنگ در یک دنیای مجازی با هم تعامل داشته

^۱Virtual reality

باشند.

محیط های مجازی در کاربردهای مختلفی می توانند مورد استفاده قرار بگیرند. به طور مثال در یک کاربرد نظامی، خلبان ها و راننده های تانک ها می توانند حضور در یک محیط جنگ مجازی را برای از بین بردن یک دشمن مجازی، در کنار یکدیگر تمرین و تجربه نمایند. از این رو مدل سازی ها و شبیه سازی های نظامی حوزه ای فعال در طراحی محیط های مجازی گسترده بوده است. شبکه های ^۱SIMNET [1] و ^۲DIS [2] نمونه های مشهور این سیستم ها هستند.

محیط های مجازی گسترده ^۳بسیاری نیز مانند NPSNET [3]، ^۴PARADISE [4]، ^۵MASSIVE [5]، MiMaze [6]، ^۵DIVE [7][8]، RING [9] و BRICKNET [10] در حوزه تحقیقات علمی شکل گرفته و تکامل یافته اند. اگرچه هدف اصلی بسیاری از این ساختارها در ابتدا نمایش کارایی الگوریتم های گسترده توسعه دهندگان آنها بوده است، این سیستم ها هم اکنون به عنوان ساختارهای پایه ای برای کاربردهای متعددی مورد استفاده قرار می گیرند. البته نکته قابل توجه این است که بیشتر این سیستم ها، برای تعداد کاربران محدود و در شبکه های محلی طراحی شده اند و تدابیر لازم برای ارسال بلادرنگ پیام ها بر روی اینترنت در طراحی آنها مورد توجه اندکی بوده است.

بازی های شبکه ای را می توان به جرأت مشهورترین حوزه رشد محیط های مجازی گسترده به شمار آورد که سرمایه های کلانی را در سال های گذشته در حوزه اقتصادی اشغال نموده اند. شبکه

¹ Simulator Network

² Distributed Interactive Simulations

³ Distributed

⁴ Performance ARchitecture for Advanced Distributed Interactive Simulation Environments

⁵ Distributed Interactive Virtual Environment

های^۱ MMOG اکنون زمینه مشارکت صدها هزار نفر از کاربران در یک دنیای مجازی هستند و بازی های شبکه ای سهم قابل توجهی از ترافیک اینترنت را به خود اختصاص داده اند. براساس پیش بینی ها، بازار بازی های شبکه ای از ۴/۵ میلیارد دلار در سال ۲۰۰۶ به ۱۳ میلیارد دلار در سال ۲۰۱۲ افزایش خواهد داشت و تعداد بازیکنان بازی های شبکه ای به میزان ۵۱٪ از سال ۲۰۰۶ تا سال ۲۰۱۲ رشد خواهد داشت [11]. به عنوان نمونه بازی WoW^۲ به تنهایی هم اکنون بیش از ۱۰ میلیون مشترک و نیم میلیون بازیکن هم زمان دارد.

یک محیط مجازی را از چند جنبه اساسی می توان مورد بررسی قرار داد. نخستین جنبه آن هسته اصلی محیط و طراحی موتور کنترلی آن است. پس از آن، مهمترین بعد مهندسی در محیط مجازی تعامل میان کاربران آن است. در یک محیط مجازی اطلاعات مختلفی باید میان کاربران مبادله شوند که از جمله آنها می توان موارد زیر را نام برد:

- توصیف اشیاء و صحنه ها
- اطلاعات صدا و تصویر
- پیام های به روز رسانی^۳

توصیف اشیاء و صحنه ها، در مواردی به کار می آید که کاربر جدید یا یک شیء جدید به محیط مجازی اضافه شود. به طور مثال زمانی که یک کاربر شیء جدیدی را به محیط اضافه می کند، توصیف آن شیء باید به سایر کاربران مورد تعامل با آن شیء فرستاده شود. ورود و خروج اشیاء و کاربران در مقایسه با سایر رویداد های محیط مجازی، فرکانس زیادی ندارد و پیام های مورد ارتباط با

^۱ Massively Multiplayer Online Games

^۲ World of Warcraft

^۳ Update message

آنها شرایط تأخیر کم را نیاز ندارند.

اطلاعات صدا و تصویر نیز ویژگی های شناخته شده ای دارند و تحقیقات زیادی بر روی چگونگی انتقال آنها انجام شده است. به طور مثال پروتکل انتقال^۱ RTP برای ارسال بلادرنگ صدا و تصویر پیشنهاد شده است.

آخرین گروه اطلاعات منتقل شده میان کاربران محیط های مجازی، پیام های به روز رسانی می باشند. هر کاربر در این دنیای مجازی با هویتی نشان داده می شود که در اصطلاح به آن اوتار^۲ گفته می شود. اوتارها می توانند با ورودی هایی که از کاربران می گیرند، محیط مجازی را تغییر دهند و این تغییر وضعیت از طریق پیام هایی به اطلاع سایر کسانی که در دنیای مجازی شریک هستند می رسد تا تمام کاربران در گذر زمان یک احساس مشترک از محیط مجازی داشته باشند. پیام های به روزرسانی به عنوان نتیجه ای از تعامل کاربران در محیط مجازی ایجاد می شوند. برای مثال زمانی که کاربری یک شیء را حرکت می دهد، پیام به روز رسانی محتوی مختصات و جهت حرکت شیء به تمام کاربران سیستم ارسال می شود و نمایش شیء برای کاربران مختلف بر مبنای این توصیف جدید تغییر می کند. برای کاهش حجم پیام های به روز رسانی، ارسال انتخابی اطلاعات به کاربران با تعریف ناحیه مورد علاقه برای هر کاربر مدیریت می شود تا از ارسال همگانی پیام های به روز رسانی به نقاط غیر ضروری پرهیز شود.

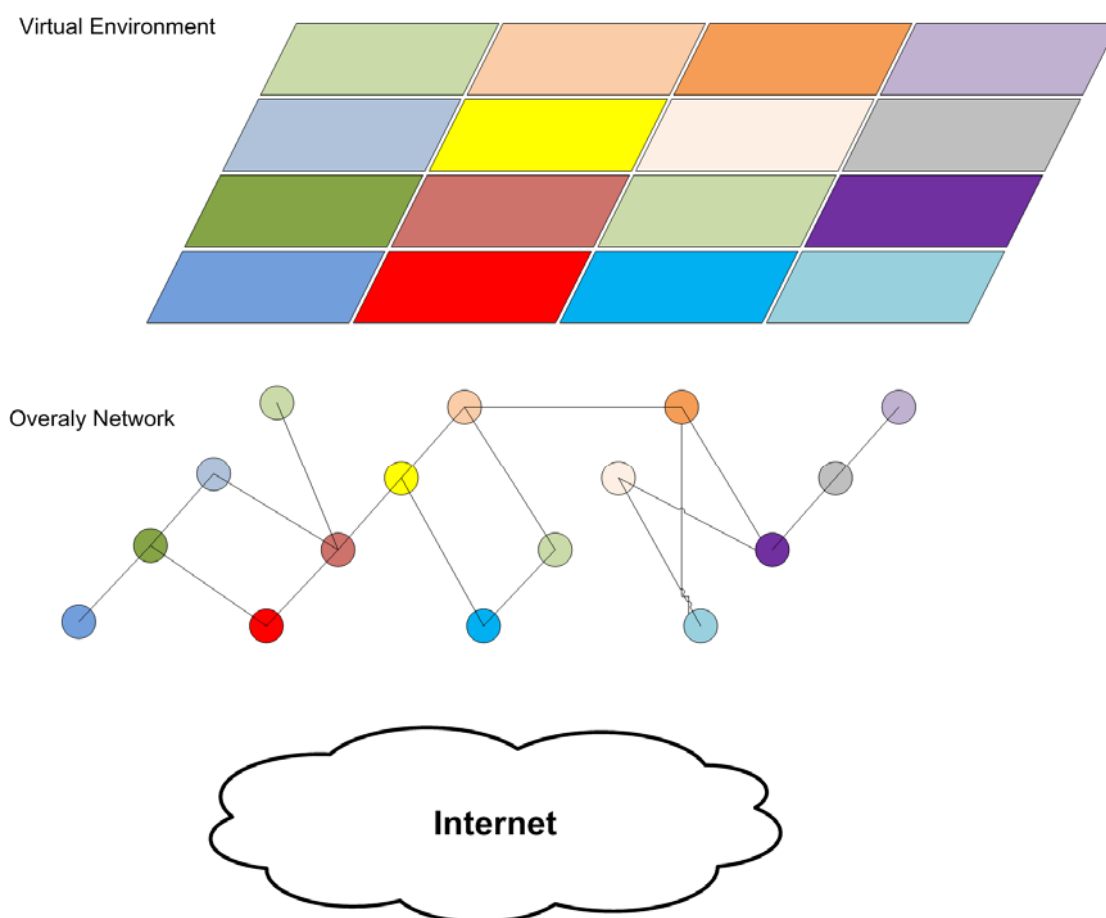
محیط های مجازی یکی از نمونه های مشهور سیستم های بزرگ با قسمت های گسترده هستند که عملکرد مناسب آنها منوط به تعامل مناسب این قسمت ها است. چنین تعاملی نیاز به یک زیر ساخت ارتباطی را ضروری می سازد. از سوی دیگر رشد و گسترش اینترنت در سال های اخیر، آن را به

^۱ Real time transport protocol

^۲ Avatar

یک شبکه زیر ساخت عمومی با دسترسی آسان تبدیل نموده است که قابلیت ارایه سرویس های با کیفیت قابل قبول و قیمت مناسب را فراهم می کند. دسترسی آسان به چنین ساختاری بسیاری از سیستم های گسترده را ترغیب به بهره گیری از آن به جای استفاده از زیر ساخت های اختصاصی نموده است. در این ساختارها انتقال پیام های به روز رسانی بر روی شبکه ای که نقاط محیط مجازی بر روی اینترنت تشکیل می دهند، انجام می شود که به این شبکه در اصطلاح شبکه overlay گفته می شود. شبکه ای که در این رساله الگوریتم های مسیریابی بر روی آن مورد بررسی قرار می گیرند، این شبکه overlay خواهد بود. شکل ۱-۱ این ساختار سه لایه ای تبادل پیام ها را در محیط مجازی نشان می دهد.

موانع متعددی بر سر پیاده سازی سیستم های گسترده بر روی اینترنت وجود دارد. به طور مثال



شکل ۱-۱ ساختار تبادل پیام ها در محیط مجازی

اگر کاربران بخواهند از خانه هایشان با هم در یک دنیای مجازی شریک شوند، تمام اطلاعات لازم برای این تعامل باید بتواند از پهنای باند محدود مودم های خانگی عبور نماید. همچنین داشتن یک شبکه گسترده که روی اینترنت کار می کند بدان معناست که پیام های کاربران در حین انتقال دچار مقادیر قابل توجهی تأخیر و لغزش^۱ خواهد شد که می تواند تعاملات بلادرنگ آنها را به میزان قابل توجهی تحت تأثیر قرار دهد. علاوه بر آن کاربرانی که بر روی سیستم های متنوع با سخت افزارها و نرم افزارهای متفاوت کار می کنند برای تعامل با یکدیگر نیاز به سیستمی یکپارچه دارند که با این تنوع سازگار باشد.

برای تبادل اطلاعات در محیط مجازی سه گونه ساختار قابل تعریف است. در ساختار سرویس مشترک- سرور^۲، توصیف محیط مجازی و اجزاء آن در یک نقطه واحد که سرور است قرار می گیرد و سرور مسئول دریافت و پردازش وضعیت تمام کاربران، محاسبه وضعیت جدید اشیاء محیط مجازی و اعلام وضعیت جدید به تمام کاربران است. در چنین شرایطی مدیریت تمام حالت ها بر عهده سرور است و با توجه به تمرکز تمام اطلاعات در سرور، مسئله نمایش و مدیریت اطلاعات پیچیدگی خاصی ندارد. زیرا سرور با داشتن اطلاعات محیط مجازی و کاربران آن، می تواند به آسانی در مورد ارسال وضعیت اشیاء مختلف محیط مجازی به کاربران تصمیم بگیرد. در ساختار مشترک- سرور، زیاد شدن تعداد کاربران منجر به محدودیت ارتباط و افزایش حجم پردازش سرورها می شود. در ساختارهایی که نیاز به پاسخگویی همزمان به تعداد قابل توجهی کاربر وجود دارد، مسأله قابلیت توسعه^۳ بسیار اهمیت پیدا می کند. برای حل این مشکل ساختارهای ترکیبی به عنوان یک راه حل مطرح شده اند که در آنها فضای محیط مجازی به چند قسمت تقسیم می شود و وظیفه مدیریت هر قسمت بر عهده یک سرور

^۱ Jitter

^۲ Client -Server

^۳ Scalability

قرار می گیرد. در این ساختارها، چالش اصلی مدیریت انتقال پیام ها هنگام جابجایی کاربران میان ناحیه هایی است که توسط سرورهای مختلف کنترل می شوند. علاوه بر آن مسأله محدودیت ظرفیت در هر ناحیه بر قابلیت توسعه پذیری تأثیر منفی می گذارد.

رشد سریع ابعاد و کاربران محیط های مجازی و نیاز به استفاده از ساختارهای قابل توسعه برای پشتیبانی کاربران بسیار زیاد از یک سو، و دسترسی به اینترنت به عنوان یک شبکه ارتباطی عمومی به عنوان یک ساختار عمومی از سوی دیگر، طراحان محیط های مجازی را به سمت استفاده از ساختارهای گسترده و عمومی برای کنترل و ذخیره سازی اطلاعات به عنوان جایگزین برای منابع متمرکز و اختصاصی سوق داد. در ساختار نظیر به نظیر^۱ هر عضو می تواند به صورت مستقیم با اعضاء دیگر ارتباط برقرار کند. بنابراین به جای انتقال پیام های از طریق یک سرور مرکزی و ایجاد گلوگاه در آن، کاربران شبکه به صورت مستقیم با یکدیگر ارتباط برقرار می کنند. برقراری ارتباطات مستقیم و بدون عبور از واسطه سرور مرکزی میان کاربران در شبکه های نظیر به نظیر باعث رفع نیاز به پهنای باند بالا در سرورها می شود. از این رو، چنین مدلی برای مشتریانی که می خواهند حجم زیادی از اطلاعات را با هم مبادله نمایند بسیار مناسب است. محور اصلی این رساله طراحی ساختارهای گسترده برای محیط های مجازی با کاربران بسیار زیاد است که در فصول آینده به تفصیل به آن خواهیم پرداخت ولی پیش از آن اندکی در مورد انگیزه های انتخاب این مسأله، چالش های آن و رویکرد حل آن صحبت خواهیم نمود.

۱.۱. انگیزه های تحقیق

در سال های اخیر رشد بسیار زیادی در حوزه محیط های مجازی چند کاربره خصوصاً بازی

^۱ Peer to Peer (P2P)

های شبکه ای اتفاق افتاده است. این رشد بی سابقه، توسعه دهندگان این صنعت را به سمت راه اندازی ساختارهایی با کاربران بسیار زیاد سوق داده است. این در حالی است که ریسک ها، پیچیدگی ها و هزینه های اولیه راه اندازی سرویس یک بازی شبکه ای با کاربران بسیار زیاد، عامل بازدارنده ای در مسیر پیاده سازی این ساختارها بوده است. از این رو ایده های به کار گیری سیستم های با قابلیت توسعه که امکان تطابق با نیازهای احتمالی آینده را بدون نیاز به پیش بینی و سرمایه گذاری اولیه داشته باشند، بسیار مطلوب به نظر می رسد. ساختارهای قابل توسعه ساختارهایی هستند که بتوانند افزایش قابل توجه کاربران را بدون نیاز به ایجاد تغییرات اساسی در طراحی اولیه پشتیبانی کنند.

بیشتر معماری های فعلی پیشنهاد شده برای محیط های مجازی، معماری های متمرکز و بر مبنای ساختار مشترک- سرور است. معماری های متمرکز قابلیت توسعه زیادی ندارند و افزایش تعداد کاربران آنها منوط به افزایش قدرت سرورهای شبکه است. درمقابل، ساختارهای نظیر به نظیر دست یابی به سیستم های قابل توسعه را تسهیل می کنند. از این رو هدف از این پروژه طراحی یک معماری گسترده و نظیر به نظیر برای یک محیط همکاری مجازی است به طوری که این ساختار در حضور تعداد زیادی از کاربران و با استفاده از شبکه اینترنت بتواند پیام های به روز رسانی یک محیط مجازی را در کمترین زمان و با کمترین هزینه منتقل نماید. بنابراین معیار اصلی سنجش کارایی در طراحی پیشنهادی، کمینه سازی تأخیر با در نظر گرفتن میزان تقاضای ترافیک مسیرها می باشد. در فصل سوم مدل مسأله و معیارهای سنجش کارایی به صورت ریاضی و به تفصیل شرح داده خواهد شد.

۲.۱. چالش ها

یکی از مهم ترین چالش ها در طراحی محیط های مجازی گسترده با کاربران بسیار زیاد تبادل

مناسب حالت های محیط مجازی هنگام تغییرات است به طوری که همه کاربران درک یکسانی از محیط مجازی داشته باشند. یک سیستم نظیر به نظیر با ابعاد بزرگ، با مساله آدرس دهی و جستجوی اطلاعات توزیع شده، روبرو است. چنین سیستمی به یک سرویس تشخیص مرجع^۱ نیاز دارد تا به کمک آن موقعیت اطلاعات (مانند نام ها و ...) را پیدا کند. محیط های مجازی گسترده از سرویس شبکه اینترنت به منظور تبادل اطلاعات میان نقاط خود استفاده می کنند.

هرچه افزایش دینامیک تغییرات در محیط مجازی بیشتر باشد نیاز به تبادل اطلاعات بیشتر جهت به روز رسانی حالت ها را ایجاد می کند. علاوه بر آن افزایش تعداد کاربران یک محیط مجازی نیاز به تبادل اطلاعات را برای هر کاربر افزایش می دهد. از آنجا که پهنای باند در دسترس کاربران خانگی در حالت کلی محدود می باشد، یکی از چالش های طراحی سیستم مورد نظر، مدیریت پهنای باند در سمت کاربران است به صورتی که این محدودیت تعاملات بلادرنگ آنها را تحت تأثیر قرار ندهد. پارامتر دیگر محدود کننده انتقال بلادرنگ پیام ها تأخیر ارسال در شبکه است که کمینه سازی آن یکی از بزرگترین مشکلات طراحی محیط های مجازی بر روی شبکه اینترنت به شمار می رود. تأخیر ارسال جزئی جدا ناشدنی از تمامی سیستم های مخابراتی به شمار می رود. گرچه قسمتی از این تأخیر به واسطه زمان انتشار سیگنال ها در کانال های مخابراتی ایجاد می شود، ولی در شبکه های عمومی مانند اینترنت بخش عمده این تأخیر ناشی از زمان های انتظار بسته ها در نقاط میانی ارسال مانند مسیریاب ها است. تأخیر ناشی از ارسال بسته ها روی شبکه اینترنت در حالت کلی تصادفی و خارج از حوزه کنترل طراحان محیط مجازی است ولی در عین حال تلاش های بسیاری برای طراحی ساختارهای محیط های مجازی صورت گرفته است، که هدف آنها گرفتن بهترین شکل ممکن سرویس از شبکه اینترنت بوده است.

^۱ Referencing

۳.۱. نقاط اصلی نو آوری

در این رساله ما روش های گسترده جدیدی را به منظور مسیریابی، کنترل و مدیریت توزیع پیام های به روز رسانی در محیط های مجازی با کاربران بسیار زیاد ارایه می دهیم. ساختارهای پیشنهادی گسترده علاوه بر قابلیت توسعه، در شرایط پیوستن و جدا شدن مداوم کاربران و همچنین تغییرات غیر قابل پیش بینی در شبکه زیرین (اینترنت) قادر به حفظ پایداری خود هستند.

اصول طراحی های پیشنهادی بر دو هدف اساسی استوار بوده است: نخستین هدف دستیابی به یک سیستم مرجع یابی^۱ به منظور دسترسی سریع به اطلاعات توزیع شده در شبکه بدون وجود یک پایگاه آدرس دهی مرکزی اطلاعات بوده است که در آن با کمک دسته بندی هوشمندانه اطلاعات، الگوریتم های جستجو و مسیریابی به شکل قابل توجهی تسریع می شود. در این رساله ما چهار خانواده مختلف از ساختارها را پیشنهاد خواهیم کرد و جزئیات طراحی آنها را برای به کار گیری در محیط های مجازی گسترده بررسی خواهیم نمود. نخستین گروه از ساختارهای گسترده پیشنهادی بر مبنای DHT^۲ است. گرچه DHT [12] در حوزه شبکه های تقسیم فایل بسیار مشهور می باشد، استفاده از آن در کاربردهای بلادرنگ و حساس به تأخیر و برای پیام های کوتاه به جای فایل های بزرگ، یک حوزه جدید و با مسایل متعدد است که ما راه کارهایی را در این حوزه پیشنهاد خواهیم داد. پس از آن ما خانواده جدیدی از ساختارها را بر مبنای منحنی هیلبرت [13] مورد بحث قرار می دهیم که در آن مسیر یابی بر روی درخت متناظر با منحنی هیلبرت انجام می شود. این ساختار به دلیل حفظ همسایگی های محلی در تبادل پیام ها در کاربرد محیط های مجازی بسیار کارآمد است. پس از آن، ساختار درخت

^۱ Indexing

^۲ Distributed Hash Table

^۱ KD [14] مورد بررسی قرار خواهد گرفت که ایده اصلی آن به کارگیری الگوریتم جستجوی سریع درخت KD در پایگاه های داده بزرگ برای مسیریابی بهینه در شبکه های بزرگ محیط های مجازی است. در نهایت آخرین ساختار پیشنهادی، ساختار گسترده هندسی است که مزیت برجسته آن ارایه سرویس مسیریابی به نقاط فضای محیط مجازی به جای آدرس ها و یا نام آنها است. چنین سرویسی طراحی ساختارهای گسترده تبادل پیام را بسیار آسان خواهد نمود. در فصول بعد این روش ها به تفصیل مورد بحث و بررسی قرار خواهند گرفت.

۴.۱. ساختار کلی رساله

ساختار کلی این رساله به ۷ فصل تقسیم شده است. در فصل دوم کلیات پایه ای موضوع و تحقیقات مرتبط انجام شده را شرح می دهیم و فصل ۳ شامل تعریف کاملی از مسأله و نمایی کلی از آن است. پس از آن در فصول متوالی راه کارهای مختلف پیشنهادی ارایه خواهد شد. فصل ۴ شامل راه حل پیش نهاد شده بر مبنای ساختار DHT است. در فصل پنجم ساختار پیشنهادی بر مبنای منحنی هیلبرت را مطرح خواهیم نمود. در فصل ششم ساختار پیشنهادی بر مبنای درخت KD را شرح می دهیم. فصل هفتم ارایه ساختار پیشنهادی بر مبنای مسیر یابی هندسی است و در نهایت آخرین فصل نتیجه گیری و مروری بر کارهای آینده خواهد بود

^۱ KD tree

فصل دوم

مروری بر کلیات و تحقیقات مرتبط

ساختارهای شبکه در محیط های مجازی

مدیریت ناحیه مورد علاقه

یک محیط مجازی شبکه ای با ابعاد بزرگ باید امکان تعامل کاربران زیادی از حدود ۱۰۰۰۰ تا ۱۰۰۰۰۰ کاربر را فراهم آورد. علاوه بر آن محیط مجازی باید احساس مشترک از زمان و مکان را برای کاربران ایجاد نماید. این بدان معنی است که همه عضوهای یک محیط مجازی باید قابلیت تعامل بلادرنگ با یکدیگر را داشته باشند. همچنین هر شرکت کننده باید از تعاملات سایرین در زمان رخ دادن آنها مطلع شود.

برای کاربران یک محیط مجازی سخت افزار های محاسباتی و خصوصا پهنای باند ارتباطی محدود می باشد. هدف در طراحی یک محیط مجازی آن است که با بهره گیری از تکنولوژی ها و روش های مناسب بهترین احساس تعامل بلادرنگ برای کاربران فراهم شود. محیط های مجازی را از نظر ساختار ارتباط شبکه ای می توان به سه گروه اصلی دسته بندی نمود.

هرکدام از این ساختارها مزایا و معایب خاص خود را دارند که در زیر فصل بعدی به آنها خواهیم پرداخت.

۱.۲. ساختارهای شبکه در محیط های مجازی

۱.۱.۲. ساختار مشترک - سرور

در این ساختار یک یا چند سرور وظیفه انتقال پیام های به روز رسانی را میان نقاط محیط مجازی بر عهده دارند. بنابراین ارتباط مستقیم میان نقاط وجود ندارد و سرورها محل انتقال پیام ها هستند. اگر در یک محیط مجازی N نقطه ای که بر مبنای مکانیزم مشترک-سرور عمل می کند، هر عضو تغییراتی را با نرخ P در محیط مشترک ایجاد کند، ارسال در هر نقطه با نرخ P انجام می شود و با نرخ $(N-1)P$ پیام های مربوط به سایرین را از سرور دریافت می کند. در چنین شرایطی سرور باید با نرخ NP دریافت و با نرخ $N(N-1)P$ ارسال انجام دهد. بنابراین با افزایش اعضا محیط مجازی، پهنای باند مورد نیاز سرور با نرخ قابل توجه زیاد می شود. از طرف دیگر، از آنجایی که نیازهای بلادرنگ نیز باید در ارسال این حجم زیاد اطلاعات تأمین شود، مسأله تبادل پیام ها بسیار دشوار است.

۲.۱.۲. ساختار نظیر به نظیر

پیشرفت های قابل توجه در حوزه کامپیوترهای شخصی که منجر به ارزان شدن و قدرتمند شدن بیش از پیش آنها شده است از یک سو و افزایش رو به رشد تعداد کاربران شبکه های مجازی گسترده از سوی دیگر، پشتیبانی مرکزی چنین سیستم هایی را دشوار و نیازمند به داشتن سرورهای بسیار گران قیمت می سازد. شبکه های نظیر به نظیر راه کاری برای گریز از این سرورهای گران قیمت هستند. هدف شبکه های نظیر به نظیر استفاده از منابع شبکه موجود در لبه های آن یعنی سمت کاربران است. شبکه های نظیر به نظیر سعی می کنند از منابع مختلف مانند ظرفیت حافظه، توان پردازش و اطلاعات کاربران مختلف خود بهره بگیرند. البته بهره گیری از منابع گسترده در شبکه به معنای کار کردن

در یک محیط ناپایدار از لحاظ اتصال است که در آن، هر عضو شبکه ممکن است در هر لحظه شبکه را ترک نماید.

انگیزه اصلی در شکل گیری شبکه های نظیر به نظیر، افزایش تعداد نقاط شبکه، رفع نیاز به پهنای باند مورد نیاز برای ارتباط نقاط از طریق یک نقطه مرکزی، افزایش توان محاسباتی و فضای ذخیره اطلاعات در مجموعه و انتقال منابع اطلاعات به نقاط انتهایی شبکه برای رفع مسأله نقطه ضعف مرکزی است. علاوه بر آن ساختارهای نظیر به نظیر دارای ویژگی های قابلیت انعطاف و مقاومت در مقابل خطاها هستند زیرا آنها می توانند در صورت لزوم کپی هایی از اطلاعات را نگه داری کنند و اطلاعات را بر روی چندین نقطه پخش کنند.

شبکه های نظیر به نظیر می توانند برای مقاصد زیر مورد استفاده قرار بگیرند:

- تقسیم فایل ها میان کاربران که مثالی از این شبکه ها Napster، Fast Track و Gnutella هستند.

- شبکه های محاسباتی گسترده مانند SETI@HOME

- ایجاد محیط های همکاری مانند Groove، myJXTA و Hive

انقلاب شبکه های نظیر به نظیر تقسیم فایل در سال ۱۹۹۹ با شهرت Napster به عنوان یک شبکه بسیار بزرگ اشتراک فایل های موسیقی آغاز شد و هم اکنون علی رغم تلاش های بسیار^۱ RIAA ترافیک نظیر به نظیر حجم زیادی از اطلاعات منتقل شده بر روی اینترنت را تشکیل می دهد.

در نسل نخست شبکه های نظیر به نظیر تقسیم فایل مانند Napster، زمانی که کاربران به دنبال یک موزیک خاص می گردند، ساختار به صورت مشترک - سرور است و مشترکین برای پیدا کردن

^۱ Recording Industry Association of America

مکان یک فایل خاص درخواست خود را به سرور می فرستند. ولی هنگام رد و بدل کردن فایل ها، Napster به عنوان یک شبکه نظیر به نظیر عمل می کند. فرار از تهدیدات قانونی و حق کپی برداری فایل های مورد اشتراک، نسل بعدی سیستم های نظیر به نظیر مانند Gnutella [16] و Freenet [14] را به استفاده از ساختارهای کاملاً گسترده سوق داد. در چنین ساختاری نقاط شبکه با همکاری یکدیگر یک شبکه overlay تشکیل می دهند که وظیفه جستجو و مسیریابی اطلاعات را به صورت گسترده انجام می دهد. جستجوی اطلاعات بدون وجود یک نقطه مرکزی چالش بزرگ این شبکه ها است. Gnutella برای رفع این مشکل از ارسال سیل آسای¹ درخواست های جستجو به تمام شبکه استفاده می کند. همه پخشی درخواست های جستجو منابع زیادی را در شبکه زیرین اشغال می کند و معمولاً زمان جستجوی بالایی را در پی دارد و از آنجا که جستجو در عمق خاصی باید متوقف شود، ممکن است به نتیجه مطلوب نرسد.

شبکه های تقسیم فایل مانند CAN [14]، Pastry [18]، Kademlia [19]، Chord [20] و Tapestry [21]، نسل سوم سیستم های نظیر به نظیر اشتراک فایل هستند که برای آدرس دهی اطلاعات خود از DHT استفاده می کنند. در فصل آینده اصول عملکرد شبکه های DHT را شرح خواهیم داد. در شبکه های CAN، Chord و Pastry با اندازه N (دارای N نقطه مستقل) هر نقطه یک جدول مسیریابی با ابعاد $O(\log(N))$ دارد. با وجود آنکه این سیستم ها روش های متفاوتی را برای دسته بندی اطلاعات و حفظ آنها هنگام پیوستن و جدا شدن نقاط از شبکه به کار می برند، ولی کارایی و قابلیت توسعه آنها مشابه می باشد. در چنین ساختارهایی پاسخ به یک درخواست به تعداد $O(\log(N))$ مرحله نیاز دارد. از آنجا که تأخیر ایجاد شده در هر مرحله به فاصله تصادفی نقاط در شبکه اینترنت بستگی

¹ Flooding

داشته و به طور مستقیم قابل کنترل نمی باشد، این مکانیزم جستجو در حالت کلی برای کاربردهای حساس به تأخیر مناسب نمی باشد.

یکی دیگر از زمینه های استفاده از شبکه های نظیر به نظیر محاسبات گسترده است که در آن از منابع گروهی از کاربران برای انجام کاری به صورت سریعتر و در زمان کمتر بهره گرفته می شود. به عبارت دیگر محاسبات گسترده شکستن یک کار بزرگ به واحد های کوچکتر و تقسیم آنها میان افراد مختلف برای انجام کار در زمان کمتر و منطقی تر است. بدین ترتیب مسایلی که شامل پردازش حجم بزرگی از اطلاعات هستند می توانند با زمان و هزینه کمتری حل شوند. به طور مثال می توان به SETI@HOME اشاره نمود که هدف آن استفاده از توان کامپیوترهای خانگی برای یافتن سیگنال های نشان دهنده حیات بود. دسته دیگر نرم افزارهایی که از محاسبات گسترده استفاده می کنند، مسایل شبیه سازی با خروجی های زیاد هستند. به طور مثال FOLDING@HOME یک نرم افزار است که در آن گروه زیادی از نقاط هر یک دنباله ای را برای folding پروتئین شبیه سازی می کنند.

استفاده از شبکه های نظیر به نظیر در کاربردهای نیازمند به تعاملات بلادرنگ حوزه ای جدید در کاربردهای این حوزه به شمار می رود. شبکه های مجازی گسترده نمونه بارزی از این کاربرد هستند. Silopsis [22] و QuON [23] نمونه ای از ساختارهای نظیر به نظیر برای محیط های مجازی هستند.

ساختارهای نظیر به نظیر با هدف پشتیبانی قابلیت توسعه و ایجاد یک محیط مجازی یکپارچه مورد توجه قرار گرفتند. در ساختارهای نظیر به نظیر تمام عضوهای شبکه در موقعیت یکسان قرار دارند و می توانند به صورت مستقیم با سایرین ارتباط برقرار کنند. علاوه بر آن هیچ نقطه مرکزی که عملکرد نقاط وابسته به آن باشد در شبکه وجود ندارد. در یک محیط مجازی نظیر به نظیر، اطلاعات محیط در میان تمام نقاط توزیع می شود و هیچ توصیف مرکزی از محیط مجازی در یک نقطه نگه داری نمی

شود. بنابراین هر یک از کاربران مسئول به روز رسانی حالت قسمتی از اشیاء برای تمام کاربران دیگر هستند. در چنین شرایطی همگرایی و سنکرون نگاه داشتن حالت ها میان کاربران بسیار دشوار است. علاوه بر آن، استفاده از ساختار نظیر به نظیر سایر پیچیدگی های کنترل یک سیستم گسترده و مدیریت مسایل امنیتی و حفاظت اطلاعات شخصی را در بر دارد، ولی در عین حال مزایای متعددی در استفاده از آن وجود دارد که این چالش ها را توجیه می کند:

- کاهش قابل توجه در هزینه سخت افزار و شبکه
- عدم نیاز به پیش بینی گسترش های آینده
- امکان کاهش تأخیر با مسیریابی پیام ها میان کاربران و حذف تأخیر ناشی از ارسال پیام ها به سرور و تأخیر پردازش آنها در سرور
- عدم اتکا به یک نقطه پشتیبانی مرکزی

۳.۱.۲. ساختار ترکیبی

چالش های مدیریت گسترده سیستم های نظیر به نظیر و مسایل توسعه پذیری از یک سو و افزایش نیاز به سرورهای قدرتمند در سیستم های کنترل مرکزی از سوی دیگر، استفاده از سیستم های دیگری را متداول نموده که ترکیبی از دو ساختار پیش هستند. برای حل مشکل اشباع توان سرور در این ساختارها از چندین سرور مختلف به جای یک سرور استفاده می شود. به طور مثال در [24] تعدادی سرور برای مدیریت یک بازی با کاربران زیاد به کار گرفته می شود که هر مشترک هنگام ورود به محیط مجازی، سرور مناسب (به طور مثال نزدیک ترین سرور از لحاظ تأخیر) را برای خود پیدا می کند و به آن متصل می شود و پس از آن با سایر سرورها ارتباطی ندارد و با سرور خود با روش مشترک-سرور ارتباط برقرار می کند. سرورها با روش نظیر به نظیر با یکدیگر ارتباط دارند و یک درخت چند پختشی

برای تبادل اطلاعات لازم تشکیل می دهند. نکته قابل توجه آن است که در این ساختار نیز مسایلی مانند تعادل بار میان سرورهای نواحی و بهینه کردن تأخیر وجود دارد [25].

۲.۲. مدیریت ناحیه مورد علاقه^۱

به منظور ایجاد احساس مشترک حضور در یک مکان واحد برای کاربران یک محیط مجازی، هر کاربر باید یک نسخه محلی از وضعیت فعلی محیط مجازی را نگه داری کند و این نسخه محلی باید در جریان تحولات محیط مجازی به صورت بلادرنگ به روز رسانی شود. بنابراین هر کاربر پس از هر تعامل باید وضعیت جدید حاصل را به تمامی کاربرانی که از آن تعامل تأثیر می پذیرند گزارش کند. ساده ترین راه حل برای این مسأله ارسال یک کپی جدید از وضعیت به تمامی کاربران به صورت همه پخشی است. از آنجا که دنیای مجازی ممکن است بسیار بزرگ و حاوی اطلاعات زیادی باشد، هر اوتار برای تعامل با محیط تنها به دانستن زیر مجموعه کوچکی از این اطلاعات نیازمند است. بنابراین روش همه پخشی با وجود سادگی، نیاز به ارسال و پردازش پیام های متعدد غیر ضروری را به وجود می آورد که مشکلات بزرگی در راه توسعه پذیری سیستم به وجود می آورد. یکی از بهترین راه حل ها در ارسال بهینه پیام های به روز رسانی آن است که به جای ارسال همه پخشی پیام ها، به هر کاربر تنها پیام های مرتبط به اوتار آن ارسال شود. مدیریت ناحیه مورد علاقه فرآیند مشخص نمودن حوزه اطلاعات مرتبط به اوتارهای یک محیط مجازی است. در این ساختار فضا به واحدهای کوچکتری تقسیم می شود که در اصطلاح به آنها نواحی مجازی و یا اتاق های مجازی گفته می شود و اعضای محیط مجازی پیام های خود را به جای ارسال به تمام اعضای محیط مجازی تنها به اتاق هایی که در ناحیه مورد علاقه آنها قرار دارند، ارسال نمایند.

^۱ Area of Interest Management (AoIM)

ایده اصلی ناحیه مورد علاقه، فیلتر نمودن پیام‌ها بر مبنای محتوای آنهاست به طوری که پیام‌ها به نقاطی که نیاز به دانستن اطلاعات آنها ندارند ارسال نشوند. مدیریت ناحیه مورد علاقه می‌تواند پهنای باند مورد نیاز برای تبادل پیام‌ها را به ویژه در ساختارهای با کاربران زیاد تا حد قابل توجهی کاهش دهد.

اطلاعات مورد علاقه یک کاربر با ادراک اوتار آن مرتبط است. از آنجا که اوتارها سرعت و قدرت حسی محدودی دارند، این اطلاعات قسمت کوچکی از اطلاعات کلی یک محیط مجازی وسیع را شامل خواهد شد. بنابراین مدیریت ناحیه مورد علاقه می‌تواند حجم اطلاعات مبادله شده در یک محیط مجازی را به میزان قابل توجهی کاهش دهد. ولی از سوی دیگر در ساختارهای گسترده مدیریت فیلتر کردن اطلاعات می‌تواند چالش‌انگیز باشد. ساختارهای مدیریت علاقه در محیط‌های مجازی به سه گروه اصلی تقسیم می‌شوند: در این قسمت به ساختارهای متداول مدیریت ناحیه مورد علاقه در محیط‌های مجازی می‌پردازیم [27] [26].

۱.۲.۲. مدل فضایی^۱

در این مدل که در اصطلاح به آن مدل Aura-Nimbus [29][28] یا مدل تعامل دو جانبه [27] گفته می‌شود، تعامل اوتارها بر مبنای موقعیت فضایی و فاصله آنها در محیط مجازی تعریف می‌شود. Aura در اصطلاح به ناحیه در برگیرنده یک شیء گفته می‌شود و Nimbus آگاهی دو جانبه اشیاء از یکدیگر را معین می‌کند. به عبارت دیگر دو شیء A و B در صورتی با یکدیگر تعامل می‌کنند که تقاطعی میان Aura های آنها اتفاق افتاده باشد و زمانی که شیء A در Nimbus شیء B قرار می‌گیرد باید از وضعیت‌ها و حالت‌های آن مطلع باشد. بنابراین اشیاء موجود در Nimbus هر شیء در محیط

^۱ Spatial Model

مجازی علاقمند به دریافت پیام های به روز رسانی در مورد حالت های آن هستند.

مدل فضایی در سیستم های مختلفی مانند MASSIVE پیاده سازی شده است. OPeN [30][31] شبکه گسترده دیگری با مدل فضایی است که بر روی شبکه ساختار یافته Chord بنا شده است که از ساختار گسترده اندیس دهی به منظور تسهیل جستجوی اشیاء استفاده می شود. در این ساختار ارتباطات میان نقاط نزدیک به صورت مستقیم انجام می شود و در سایر موارد، هر نقطه شبکه موقعیت فعلی خود را در فضای Chord ثبت می کند و پاسخ درخواست های خود را در مورد سایر نقاط را از طریق Chord جستجو می کند. برای ذخیره حالت های ماندگار نیز از یک سرور مرکزی استفاده می شود.

مسئله اصلی روش فضایی، نیاز به تبادل پیام های متعدد به روز رسانی مکان میان نقاط برای آگاهی از تداخل در Nimbus آنها می باشد که با توجه به حرکت های دائمی نقاط می تواند پیچیده باشد. برای حل این مشکل راه حل های مختلفی پیشنهاد شده است. ساختار پیشنهاد شده در [32] مدل فضایی بهینه شده ای است که در آن پیام های به روز رسانی با نرخ پایین برای تشخیص اشیاء مورد استفاده قرار می گیرند و هنگام تشخیص افزایش احتمال تعامل اشیاء نزدیک، پیام های به روز رسانی با نرخ متغیر استفاده می شوند. در نهایت هنگام تعامل اشیاء پیام های به روز رسانی با نرخ بالا ارسال می شوند. استفاده از همسایگی های Voronoi نیز روش دیگری برای کاهش پیچیدگی روش فضایی است. در این راهکار، دیاگرام Voronoi برای تخصیص همسایگی ها در محیط مجازی به کار می رود [33]-[36]. بدین ترتیب هر نقطه یک دیاگرام Voronoi میان خود و همسایگانش تشکیل می دهد و تنها با این همسایگان در دیاگرام Voronoi در ارتباط است. نکته قابل توجه آن است که با حرکت هر نقطه، همسایگی ها باید به روز رسانی شوند. همچنین با ورود یا خروج هر همسایه دیاگرام Voronoi یک بار به روز رسانی می

شود. به این منظور هر نقطه از اطلاعات خودش و همسایگان فعلیش کمک می گیرد تا نقاط تازه وارد به همسایگی اش را پیدا کند و نقاط دور شده را از همسایگیش حذف کند.

۲.۲.۲. مدل های چند پخشی در لایه کاربرد^۱

در این مدل که در اصطلاح به آن مدل ناحیه ای با مکانیزم ارسال همگانی و ثبت نام^۲ هم گفته می شود، فضای محیط مجازی به تعدادی ناحیه معمولاً ثابت تقسیم می شود و برای هر ناحیه یک گروه چند پخشی تعریف می شود. زمانی که سطح مورد علاقه یک نقطه با یک یا چند عدد از این نواحی در تداخل باشد، آن نقطه باید تمام پیام های به روزرسانی آن ناحیه ها را دریافت کند. بنابراین هر نقطه با گروه چند پخشی آن نواحی ثبت نام می کند. همچنین در صورتی که در اثر حرکت نقطه ای، سطح مورد علاقه اش تغییر کرده و دیگر با ناحیه ای که قبلاً با آن ثبت نام کرده است همپوشانی نداشته باشد، نقطه مورد نظر اشتراک خود را با آن ناحیه لغو می کند [37][38].

P2PArch [39] یک ساختار چند پخشی است که بر روی شبکه Pastry طراحی شده است و از مکانیزم چند پخشی تعریف شده در Pastry به نام Scribe [40] به منظور چند پخشی پیام ها در نواحی استفاده کرده است و از مکانیزم PAST برای نگاه داری حالت های بازی پیشنهاد داده است. SimMud [41][42] ساختار ناحیه ای دیگری است که در آن هر ناحیه به عنوان یک کانال انتشار تعریف می شود که تمام اعضای آن ناحیه، عضو آن می شوند و از SCRIBE برای انتشار پیام ها میان اعضای هر ناحیه استفاده می شود. Colyseus [43] نسخه تغییر یافته SimMud در تلاش برای بهبود تأخیر جستجو در DHT می باشد.

^۱ ALM : Application Layer Multicast

^۲ Regional Publish/Subscribe

Mediator [44] نیز ساختار دیگری است که از چند پخشی در لایه کاربرد برای نگه داری و به روز رسانی حالت های محیط مجازی استفاده می کند. ولی در کنار آن ارتباطات نظیر به نظیر مستقیم نیز بر مبنای نیاز در شرایط حساس به تأخیر استفاده می شوند.

مدل چند پخشی در قیاس با مدل فضایی از نظر پیاده سازی ساده تر است و محاسبه نواحی ارسال پیام ها پیچیدگی کمتری دارد. ولی در مقابل دقت ارسال پیام های به روز رسانی در آن کمتر و در حد اندازه نواحی است. بنابراین در شرایطی که یک نقطه تنها ممکن است به دریافت تعداد کمی از پیام های به روز رسانی یک ناحیه علاقمند باشد، برای دریافت تمام پیام های آن ناحیه ثبت نام می کند. این مسأله حجم زیادی از سرباره را در ارسال و دریافت پیام ها به سیستم تحمیل می کند.

۳.۲.۲. مدل های ترکیبی

این ساختار که به آن ساختار مبتنی بر نقطه قدرتمند^۱ نیز گفته می شود، ترکیبی از دو ساختار پیشین است که قسمتی از معایب هر ساختار را با ترکیب آنها رفع می کند. مشابه ساختار دوم در اینجا نیز محیط مجازی به تعدادی ناحیه ثابت تقسیم میشود. ولی الگوریتم مدیریت سطح مورد علاقه در هر ناحیه به صورت فضایی و بر مبنای ساختار اول بوده و وظیفه انجام آن بر عهده یک نقطه قدرتمند در آن ناحیه به عنوان مسئول ناحیه است [45][46].

P2PSE [47] نمونه ای از یک ساختار ترکیبی از تعدادی گروه است که تبادل اطلاعات درون آنها به صورت نظیر به نظیر است و در کنار آن از تعدادی سرور برای اتخاذ تصمیمات کلی استفاده می شود. Hydra [48] نیز نمونه دیگری است که در آن فضای محیط مجازی به تعدادی ناحیه افراز می شود که هر ناحیه توسط یک سرور کنترل می شود. قابل توجه است که سرور هر ناحیه باید از میان نقاط آن

^۱ Supernode- based

ناحیه انتخاب شود. همچنین یک کنترل کننده کلی نیز در سیستم وجود دارد که آدرس IP سرورهای نواحی را نگهداری می نماید. استفاده از DHT برای پیاده سازی گسترده این کنترل کننده عمومی پیشنهاد شده است.

[51] Zoned Federation of Games Servers ، [50] MetaModel ، [49] MOPAR

[52] Distributed '04 و [53] Mercury نمونه های دیگری از این سیستم های ترکیبی هستند که تبادلات درون ناحیه ای آنها به صورت مستقیم و نظیر به نظیر و تبادلات میان نواحی به کمک DHT انجام می شود. در ساختارهای پیشنهادی در [54] محیط مجازی به تعدادی ناحیه تقسیم می شود و میان نقاط شبه ساختار یافته CAN توزیع می شود که توسط آن ورود و خروج نقاط کنترل شده و مختصات آنها نگه داری می شود. هر ناحیه می تواند یک سرور داشته باشد و در صورت نیاز به یک سیستم کاملاً گسترده این سرور می تواند با یک نقطه اصلی انتخابی درون ناحیه جایگزین شود. COVER [55] نیز ساختار ترکیبی دیگری است که در آن محیط مجازی به صورت یک ساختار درخت چهارگانه تقسیم بندی شده است که هر ناحیه تعداد محدودی کاربر و یک نقطه اصلی^۱ دارد. همه پیام های به روز رسانی به نقطه اصلی ارسال می شود تا در صورت لزوم به نقاط خارجی ارسال شود هر نقطه پیام حاوی به روز رسانی خود را به همسایگانش ارسال می کند که آنها نیز این پیام را در صورت لزوم به همسایه های رده دوم باز ارسال می کنند.

۴.۲.۲. مدل مورد استفاده

روش های ارائه شده در قسمت های قبل برای مدیریت ناحیه علاقه هر کدام مزایا و معایب خاص خود را دارند. روش فضایی دارای کارایی بسیار مناسبی می باشد ولی پیچیدگی پیاده سازی از

^۱ Super node

مشکلات مهم آن است. روش چند بخشی در لایه کاربرد با وجود سادگی دارای سرباره و تأخیر زیاد در انتقال پیام هاست و روش ترکیبی از نظر تأخیر و پایداری قابل قبول است ولی مسایل جابجایی میان نواحی و تعادل بار در آنها پیچیدگی های متعددی در آن ایجاد می کند.

مدل پیشنهادی ما برای مدیریت سطح مورد علاقه بر مبنای روش فضایی می باشد که از نظر عملکرد حداقل پیام های اضافی را مبادله می کند. ولی به منظور حذف پیچیدگی های مرتبط با محاسبه تداخل میان Nimbus اشیاء، روش جدیدی پیشنهاد می شود. ایده اصلی روش جدید آن است که هر نقطه محیط مجازی پیام های به روز رسانی خود را به ناحیه Nimbus خود ارسال کند تا تمام افراد موجود در آن این پیام ها را دریافت نمایند. بدین صورت هر نقطه نیاز ندارد تا به صورت مداوم موقعیت نقاط دیگر را کنترل نماید. اجرای چنین روشی نیاز به یک سرویس مسیریابی برای ارسال پیام ها به موقعیت های مشخص در محیط مجازی دارد که در ادامه رساله به آن می پردازیم.

میان Nimbus و موقعیت یک کاربر در محیط مجازی ارتباط نزدیک وجود دارد. این بدان دلیل است که ادراک بصری از یک نقطه به صورت کلی با فاصله اوتار از آن نسبت دارد. البته قابل توجه است که دید یک اوتار همیشه با فاصله نسبت ندارد، بلکه موانع می توانند مسیر دید اوتارها را مسدود کنند. به طور مثال اوتارها قادر به دیدن از میان دیوارها نیستند در حال که اشیاء آن سوی دیوار ممکن است در فاصله نزدیک از اوتار باشند. از این رو یک مکانیزم مناسب مدیریت ناحیه علاقه باید اثر موانع را در ارسال پیام ها در نظر بگیرد [56]. با این وجود فاصله مجازی در حالت کلی معیار مناسبی برای در نظر گرفتن علاقه کاربران است. از این رو در ادامه این رساله فرض ما بر آن خواهد بود که ناحیه مورد علاقه یک کاربر در محیط مجازی فضای در بر گیرنده اطراف آن است و میزان علاقه اشیاء به فاصله آنها بستگی دارد. فصول آینده جزئیات روش های پیشنهادی را در بر خواهند داشت.

فصل سوم

تعریف و مدل سازی

پهنای باند و کیفیت پیوندهای شبکه زیرساخت

ماتریس ترافیک

ماتریس تأخیر

تابع هدف

یافتن پاسخ بهینه

هدف این رساله طراحی یک ساختار گسترده برای محیط های همکاری مجازی است به طوری که این ساختار در حضور تعداد زیادی از کاربران و با استفاده از شبکه اینترنت بتواند عملکرد مناسبی از خود نشان دهد. از این رو عمده توجه ما بر روی ارایه یک ساختار overlay مناسب بر روی شبکه اینترنت و ارایه الگوریتمهای بهینه مسیریابی بر روی آن بوده است. نخستین گام در انجام این کار، تعریف درست و دقیق از اهداف مسأله و معیارهای سنجش کارایی آنهاست.

نقاط شبکه overlay از میان تمام نقاطی که قابلیت همسایگی آنها را دارند، تعدادی را بر اساس معیارهای از پیش انتخاب شده به عنوان همسایه های خود انتخاب می کنند. به فرآیند این انتخاب در اصطلاح کنترل توپولوژی گفته می شود. اگر همسایگی های شبکه به صورت یک ماتریس اتصال نمایش داده شوند که یک بودن هر درایه آن مشخص کننده وجود یک ارتباط مستقیم (همسایگی) میان دو نقطه متناظر آن روی overlay است، هدف الگوریتم کنترل توپولوژی مشخص کردن درایه های این ماتریس است به طوری که شبکه overlay حاصل بهترین کارایی را بر طبق معیارهای از پیش تعیین شده داشته

باشد. مسأله کنترل توپولوژی با پارامترهای ورودی و تابع هدف آن تعریف می شود. در زیر فصل های بعدی جزئیات این پارامترها و تعریف کاملی از آنها آورده شده است.

۱.۳. پهنای باند و کیفیت پیوند^۱ های شبکه زیرساخت

اگر شبکه اینترنت به عنوان شبکه زیرین در نظر گرفته شود، پهنای باند مسیرها به ارتباطات نهایی^۲ سمت کاربر محدود می شود و مسیرهای هسته^۳ معمولاً به جای محدودیت های ظرفیتی، محدودیت زمانی بر روی ارسال ایجاد می کنند. کاربران یک محیط مجازی با واسط های مختلفی به شبکه اینترنت متصل هستند. بنابراین پهنای باند آنها از مقادیر ۵۶Kbps و کمتر برای کاربران dialup تا ۵۱۲Kbps و بیشتر برای کاربران سرویس های باند وسیع خانگی و بالاتر از آن برای سایر کاربران غیر خانگی که دسترسی مستقیم به پیوندهای ماهواره ای و یا نوری دارند، متغیر است. یکی از چالش های شبکه های محیط های مجازی ایجاد هماهنگی میان کاربرانی است که با پهنای باندهای مختلف به اینترنت اتصال دارند. در شبیه سازی های فصول بعدی، پهنای باند ارسال کاربران در محدوده های مختلفی تعریف شده است که کاربران خانگی متصل از طریق خط تلفن با سرعت ۵۶Kbps تا کاربران ADSL با پهنای باندهای چندین Mbps را شامل می شود.

معیارهای مختلفی برای سنجش کیفیت ارتباط پیوندهای اینترنت می تواند به کار گرفته شود که مهمترین آنها تأخیر مسیرها می باشد. در ارتباطات بلادرنگ، لغزش^۴ نیز یک پارامتر مهم در ارتباط به شمار می رود. تفاوت مهم میان تأخیر و لغزش در مسأله بهینه سازی ماهیت جمع شونده تأخیر می باشد

^۱ Link

^۲ Last mile

^۳ Core

^۴ Jitter

که کار کردن با آن را بسیار ساده تر می کند. ویژگی جمع شونده بودن تأخیر بدان معنا است که در عبور از تعدادی پیوند، تأخیر کلی مسیر معادل با مجموع تأخیر آن پیوند هاست. قابل توجه است که چنین ویژگی در مورد لغزش وجود ندارد. در عین حال، با کمینه کردن تأخیر می توان لغزش را تا حدودی با بافر کردن ترافیک و در نتیجه افزایش تأخیر، کاهش داد. بنابراین در این رساله تأخیر به عنوان پارامتر اصلی نشان دهنده کیفیت در نظر گرفته می شود.

۲.۳. ماتریس ترافیک

ماتریس ترافیک مشخص کننده میزان ترافیک جابجا شده میان هر دو نقطه شبکه می باشد. به طور معمول این ماتریس شکل همگن ندارد و نیاز به ارسال ترافیک میان برخی نقاط که در ناحیه علاقه یکدیگر قرار دارند از سایر نقاط بیشتر است.

ماتریس ترافیک در محیط مجازی تابع دو پارامتر اصلی می باشد. نخستین این پارامترها توزیع کاربران در محیط مجازی است و پارامتر دوم ارتباط تقاضای تبادل پیام های به روز رسانی با فاصله میان کاربران است. در این قسمت این دو پارامتر را به تفصیل شرح خواهیم داد.

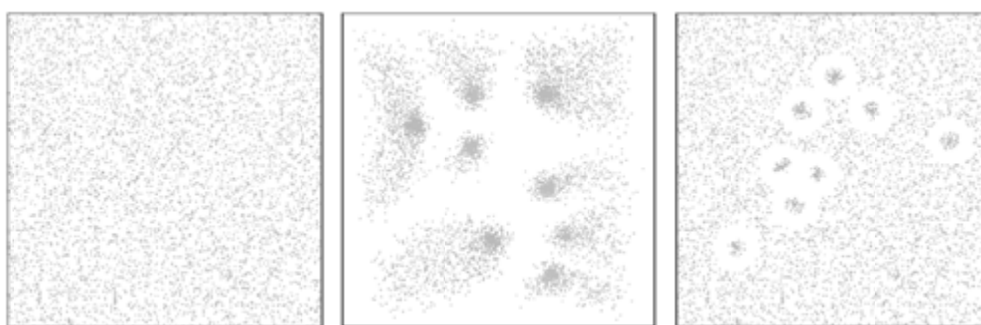
۱.۲.۳. توزیع کاربران

محیط مجازی مورد نظر ما شامل N نقطه است که N می تواند عددی بزرگ باشد که در حوزه 10^3 تا 10^5 تغییر می کند. همچنین فرض ما بر آن است که بسته به تعداد کاربران حدود 20 تا 200 ورود و خروج به شبکه در هر زمان وجود دارد.

توزیع کاربران در محیط مجازی می تواند به شکل های مختلف باشد. مدل توزیع CCP^1 [57]

¹ Changing Circular Pattern

بر مبنای اندازه گیری های انجام شده بر روی محیط مجازی CYBERWALK [58] پیشنهاد شده است. در این مدل هر نقطه حرکت دایره ای دارد که از یک مکان در محیط مجازی شروع می شود و به همان مکان منتهی خواهد شد. هر حرکت به میزان ۱۵ واحد در جهت دید اوتار است و زاویه دید در هر مرحله به میزان ۱۲ درجه تغییر می کند. علاوه بر آن جهت حرکت در هر ۴ مرحله به میزان ۱۰ درجه تغییر می کند. بنابراین اوتارها به صورت تصادفی حول صحنه ها در مسیرهای دایره ای در حرکت هستند. مدل های دیگر در حرکت کاربران در محیط مجازی، مدل های HPN و HPA هستند [59]. در مدل HP-All (HPA) فرض بر آن است که تعدادی نقطه پرازدحام^۱ در محیط مجازی وجود دارند که اوتارها برای حرکت به سمت این نقاط مجازی تمایل دارند. در محیط هایی که در آنها هدف کاربران دستیابی به منابعی مانند انرژی، غذا، اسلحه و یا رسیدن به نقاط امتیاز آور است، چنین مدلی به خوبی صدق می کند. در مدل HPN هم نقاط پرازدحام مشابه مدل HPA تعریف شده اند. ولی تفاوت در آن است که تنها اوتارهایی که در محدوده معینی در اطراف یک نقطه پرازدحام باشند به سمت آن حرکت می کنند. شکل ۱-۳ توزیع کاربران را در محیط مجازی در دستگاه دو بعدی مختصات مجازی آنها و در هر سه



(ج)

(ب)

(الف)

شکل ۱-۳ توزیع کاربران در محیط مجازی (الف) HPA (ب) HPN (ج) CCP

^۱ Hot Point

حالت ذکر شده نشان می دهد [59].

مدل مورد استفاده ما در این رساله مدل HPN است که با رفتار بیشتر محیط های مجازی معروف تطابق بیشتری دارد. علاوه بر آن ما فرض دیگری را نیز در مورد توزیع کاربران اضافه می کنیم: اوتارهای یک محیط مجازی سه بعدی بیشتر در نقاط نزدیک به صفحه $z=0$ متمرکز هستند و تراکم آنها با افزایش z کاهش می یابد.

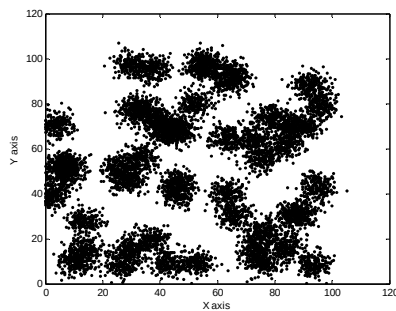
این توزیع از آنجا ناشی می شود که بیشتر اوتارهای یک فضای سه بعدی بر روی زمین قرار دارند و تعداد کمتری در بلندی ها یا وضعیت پرواز قرار می گیرند. بنابراین چگالی جمعیت با دور شدن از صفحه XY کاهش می یابد. شکل ۲-۳ توزیع کاربران را برای یک محیط مجازی با ۱۰۰۰۰ کاربر که حول ۵۰ نقطه متمرکز شده اند نشان می دهد.

۲.۲.۳. ارتباط تقاضای تبادل پیام های به روز رسانی با فاصله کاربران

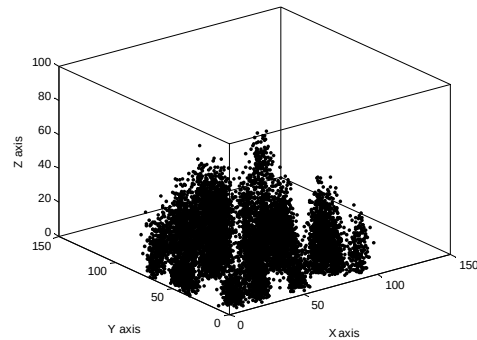
در فصل گذشته الگوریتم مدیریت ناحیه مورد علاقه را به اختصار شرح دادیم و اشاره کردیم که روش مورد استفاده ما در این رساله نسخه تغییر یافته روش فضایی است. در این روش، هر نقطه پیام های به روز رسانی خود را به ناحیه اطرافش که از آن تأثیر می پذیرند ارسال می کند. بنابراین میزان تقاضای تبادل پیام های به روز رسانی میان دو نقطه به فاصله آنها در محیط مجازی بستگی دارد و اگر این فاصله از حد معینی بیشتر شود، دو نقطه نیازی به تبادل پیام های به روز رسانی ندارند. بنابراین میزان تقاضای ارسال ترافیک از نقطه i به نقطه j با رابطه زیر قابل نمایش است.

$$d_{ij} = f(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}) \quad (۱-۳)$$

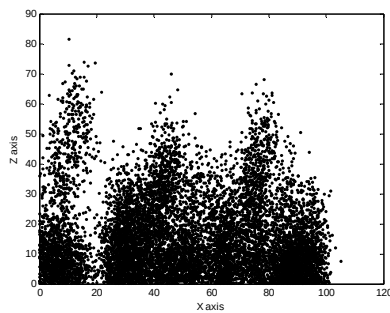
که در آن f یک تابع غیر صعودی و نامنفی تعریف می شود و اگر فاصله از حدی بیشتر شود تقاضا مساوی صفر در نظر گرفته می شود.



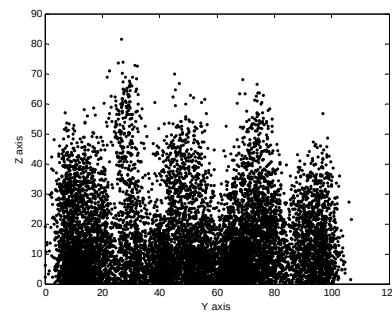
(ب)



(الف)



(د)



(ج)

شکل ۳-۲ توزیع کاربران در محیط مجازی الف) سه بعد (ب صفحه xy (ج صفحه yz (د صفحه xz

در شبیه سازی های آینده برای سهولت فرض شده است که پیام های به روز رسانی هر نقطه به کره ای در اطراف آن ارسال می شود. اگر ابعاد محیط مجازی در امتداد محورهای X, Y و Z در محدوده $[0 \ 100]$ در نظر گرفته شود، ناحیه تأثیر هر نقطه که پیام های به روز رسانی به آن ارسال می شوند عددی تصادفی در محدوده $[1 \ 3]$ در نظر گرفته می شود. همچنین نرخ ارسال پیام های به روز رسانی کاربران، متغیر از 40 Kpbs تا 1 Mbps تعریف شده است

۳.۳. ماتریس تأخیر

راه های مختلفی به منظور پیش بینی تأخیر میان نقاط در شبکه اینترنت بدون نیاز به انجام اندازه

گیری میان تمام نقاط، وجود دارد. دستگاه های مختصات سنتزی¹ یکی از روش های پیش بینی تأخیر میان نقاط شبکه اینترنت است که روش های مختلفی برای تشکیل آنها پیشنهاد شده است [63]-[60]. در دستگاه مختصات سنتزی به هر نقطه شبکه مختصاتی تخصیص داده شود به طوری که فاصله اقلیدسی میان هر دو نقطه در این دستگاه، معرف تأخیر میان آن دو نقطه در شبکه اینترنت باشد. بنابراین اگر هر نقطه مختصات سنتزی نقطه دیگری را بداند می تواند تأخیر میان خود و آن نقطه را بدون انجام هیچ اندازه گیری تخمین بزند.

دستگاه مختصات سنتزی با مشارکت گسترده نقاط شبکه ساخته می شود. محاسبه مختصات هر نقطه در دستگاه مجازی سنتزی با توجه به فاصله نقطه مورد نظر با تعدادی نقطه مرجع² با مختصات مشخص، صورت می گیرد. هر نقطه با دانستن مختصات نقاط مرجع و اندازه گیری فاصله خود از آنها، مختصات خود را به گونه ای انتخاب می کند که فاصله محاسبه شده با نقاط مرجع در دستگاه سنتزی، کمترین میزان خطا را با فاصله اندازه گیری شده با آن نقاط داشته باشد. نقاط شبکه مختصات های دستگاه سنتزی را در کنار آدرس IP در جدول مسیریابی خود نگه داری می کنند. بنابراین هر نقطه در کنار آدرس IP همسایه های خود می تواند تأخیر میان خود و آنها را تخمین بزند و از آن در تصمیم گیری های مربوط به مسیریابی استفاده نماید.

بنابراین در ادامه این رساله ما با دو نوع متفاوت از فاصله سروکار خواهیم داشت. فاصله مجازی نشان دهنده فاصله نقاط از یکدیگر در محیط مجازی است و فاصله جغرافیایی که بر مبنای مختصات سنتزی محاسبه می شود، نشان دهنده فاصله نقاط از یکدیگر بر روی شبکه اینترنت است. حرکت نقاط در محیط مجازی باعث تغییر شناسه آنها و فاصله مجازی آنها می شود، در حالی که تغییرات پیوندهای

¹ Synthetic

² Landmarks

شبکه اینترنت باعث تغییر فاصله جغرافیایی نقاط می شود.

به منظور انجام شبیه سازی ها در فصول آینده، ماتریس تأخیر شبکه اینترنت به عنوان ورودی شبیه ساز در نظر گرفته می شود. بدین ترتیب شبیه سازی توپولوژی اینترنت و مقادیر تأخیر آن که خود حوزه ای گسترده و بحث برانگیز است، خارج از دامنه شبیه سازی های این رساله می باشد.

۴.۳. تابع هدف

تعریف مسأله کنترل توپولوژی نیازمند مشخص کردن تابع هدف آن است. از آنجا که overlay به عنوان بستر انتقال پیام ها در محیط مجازی عمل می کند، ارایه کیفیت سرویس مطلوب از معیارهای اولیه در انتخاب آن است. همانطور که قبلاً اشاره نمودیم، در این رساله تأخیر به عنوان معیار اصلی در کیفیت سرویس در نظر گرفته می شود. بنابراین تابع هدف باید به صورتی تعریف شود که تأخیر کلی در شبکه حداقل شود و در می نیم کردن تأخیر مسیرها ارجحیت با مسیر هایی باشد که حجم بیشتری از ترافیک را منتقل می کنند.

توپولوژی یک شبکه overlay یک گراف جهت دار $G = (V, E)$ است که در آن $V = \{v_1, v_2, \dots, v_N\}$ مجموعه نقاط شبکه overlay و E مجموعه رابط های متصل کننده این نقاط می باشد که هر رابط e_{ij} در E اتصال از نقطه v_i به نقطه v_j را برقرار می کند. فرض بر آن است که هر رأس $v \in V$ به شبکه اینترنت اتصال دارد و اصولاً امکان برقراری ارتباط با هر رأس دیگر را دارد. برای در نظر گرفتن کیفیت سرویس پیوندها، تابع هزینه f از $E \rightarrow C$ تعریف می شود که به هر لبه گراف عددی مثبت را که معادل با هزینه عبور ترافیک از آن لبه است، نسبت می دهد.

در جریان تشکیل توپولوژی هر رأس $v \in V$ زیر مجموعه E_p از رأس ها $N(v) \subseteq V$ را به

عنوان همسایه های خود انتخاب می کند:

$$E_p = \{(u, v) | u \in V, v \in N(u)\} \quad (۲-۳)$$

خروجی الگوریتم کنترل توپولوژی یک زیر گراف سازنده^۱ از G به نام $G_p = (V, E_p)$ که G_p یک گراف پیوسته و متقارن است. نرخ انتقال پیام های به روز رسانی روی این گراف با ماتریس $P = [p_{ij}]$ توصیف می شود که p_{ij} نرخ ارسال پیام های به روز رسانی میان مبدأ i و مقصد j می باشد.

در کنار شبکه overlay تشکیل شده، الگوریتم مسیریابی F نیز تعریف می شود. این الگوریتم باید بتواند در هر نقطه میانی v_i بسته ای به مقصد v_j را به بهترین نقطه در مجموعه $V \setminus v_i$ ارسال نماید به طوری که بسته در کوتاهترین مسیر ممکن به مقصد برسد و مسیر بسته تا مقصد آن از هیچ نقطه ای دو بار عبور نکند (در مسیر حلقه ای وجود نداشته باشد). هر شبکه overlay با N نقطه، یک ماتریس همسایگی با ابعاد $N \times N$ دارد که یک بودن درایه (i, j) در آن نشانگر همسایگی نقاط i و j روی شبکه overlay است و صفر بودن آن خلاف این فرض را نشان می دهد. برای هر نقطه منبع s ، هر نقطه مقصد d و ماتریس همسایگی D برای گراف G_p ، الگوریتم مسیریابی F مسیر $X^{sd}(D)$ را میان ورودی و خروجی تعیین می کند:

$$F: D \rightarrow X^{sd}(D) \quad (۳-۳)$$

که $X^{sd}(D)$ زیر ماتریسی از D است به طوریکه $X_{ij}^{sd} = 1$ نشان دهنده استفاده از لینک میان v_j و v_i در ارسال ترافیک میان s و d است.

بسته های جابجا شده میان نقاط از دو نوع اصلی هستند. قسمت اول بسته های حامل پیام های

^۱ Spanning Subgraph

به روز رسانی هستند و قسمت دوم بسته های حامل اطلاعات کنترل برای به روز رسانی جدول های مسیریابی نقاط می باشند که به این اطلاعات در اصطلاح اطلاعات کنترل توپولوژی گفته می شود و هزینه تبادل آنها از رابطه (۴-۳) قابل محاسبه است:

$$C_i^{INF}(d_i) = \alpha_i N_i(d_i) \quad (۴-۳)$$

که d_i سطر i ام ماتریس همسایگی D است و $N_i(d_i)$ تعداد همسایگان نقطه v_i در شرایطی است که ماتریس اتصال شبکه D باشد.

α ضریبی است که برای هزینه پیام های کنترل توپولوژی در نظر گرفته می شود و در کاربردهای مختلف، مقادیر متفاوتی به آن نسبت داده می شود. در حالت کلی α حد مصالحه میان پهنای باند و سایر منابع صرف شده برای تبادل اطلاعات کنترلی و کیفیت سرویس حاصل از تبادل این اطلاعات را معین می کند.

دسته دوم اطلاعات منتقل شده روی overlay، پیام های به روز رسانی هستند. هزینه انتقال این پیام ها در شبکه برای نقطه v_i ، در رابطه (۵-۳) تعریف شده است :

$$C_i^{COM}(D) = \sum_j \sum_{m,n} (X_{mn}^{ij} c_{mn}) p_{ij} \quad (۵-۳)$$

که c_{mn} هزینه عبور پیام از پیوند میان نقاط m و n است. هزینه کلی که مجموع دو پارامتر قبلی است به صورت زیر تعریف می شود:

$$C^{TOT}(D) = \sum_i C_i^{COM}(D) + C_i^{INF}(d_i) \quad (۶-۳)$$

هدف مسأله یافتن ماتریس اتصال D به منظور کمینه کردن تابع هزینه بالاست:

$$\text{Min } C^{\text{TOT}}(D) = \sum_i C_i^{\text{COM}}(D) + C_i^{\text{INF}}(d_i) \quad (7-3)$$

در فصول آینده این ارسال روشهای گسترده مختلفی را به منظور تشکیل توپولوژی و مسیر یابی ارایه خواهیم داد که هدف آنها به کار گیری راه حل های تقریبی به جهت کم کردن تابع هزینه ذکر شده است. پیش از آنکه در فصول آتی به جزئیات این روش ها پردازیم، در قسمت باقیمانده این فصل به حل مسأله ذکر شده برای یافتن پاسخ بهینه آن در یک حالت خاص و بدون در نظر گرفتن هزینه کنترل توپولوژی می پردازیم. به این منظور ما سعی در مدل کردن این مسأله در قالب یک مسأله در تئوری گراف داشته ایم تا بتوانیم از راه حل های کلاسیک در حل مسأله مورد نظر استفاده کنیم. در چنین مدلی نقاط شبکه به عنوان نقاط گراف و پیوند های میان آنها به عنوان لبه های گراف تعریف می شوند که پهنای باند هر پیوند ظرفیت لبه متناظر در گراف را مشخص می کند. یکی از حوزه های تئوری گراف در ریاضیات، حوزه ای به عنوان جریان های شبکه^۱ است. هدف اصلی این تئوری انتخاب مناسب مسیرهای بهینه و با حداقل هزینه در یک گراف است که در آن تعدادی از نقاط به عنوان فرستنده و تعدادی به عنوان گیرنده تعریف می شوند. ساده ترین مسأله در حوزه جریان های شبکه، حالت حل استاتیک مسأله در حضور یک فرستنده و یک گیرنده است. در حالت های پیچیده تر تعداد فرستنده ها و گیرنده می تواند بیشتر باشد و مسأله به جای حالت استاتیک در حالت دینامیک حل می شود. در حل دینامیک مسأله برای پیوند های گراف علاوه بر ظرفیت که نشان دهنده پهنای باند آنهاست، پارامتر دیگری به نام تأخیر هم در نظر گرفته می شود و هدف حل مسأله مشخص کردن مسیری است که در آن میزان مشخص ترافیک بتواند در کوتاه ترین زمان منتقل شود. در این قسمت به شرح جزئیات حل مسأله با این روش خواهیم پرداخت.

^۱ Network Flows

۵.۳. یافتن پاسخ بهینه

بهینه سازی تأخیر به مفهوم پیدا کردن بهترین همسایگی هایی است که کمترین میزان تأخیر را میان نقاط یک محیط مجازی گسترده نتیجه دهد. تأخیرهای نظیر به نظیر میان گروهی از منابع و مقاصد در شبکه تعریف می شوند که زیر مجموعه ای از نقاط محیط مجازی هستند.

میان هر دو نقطه محیط مجازی که نیاز به تبادل پیام های به روز رسانی داشته باشند، تقاضایی برای ارسال ترافیک تعریف می شود. همسایگی هایی که به بهترین صورت مسیر لازم برای سرویس دهی به این تقاضاها را فراهم نمایند توپولوژی بهینه را برای شبکه تشکیل می دهند. لازم به ذکر است که مسیرهای بهینه در اصطلاح به مسیریایی گفته می شود که کمترین تأخیر را در عبور ترافیک ایجاد نمایند. در اینجا به دنبال یافتن کمترین مقدار مطلق تأخیر هستیم و بنابراین از هزینه پیام های کنترل توپولوژی صرف نظر می کنیم و هزینه ذکر شده در قبل در واقع بدون در نظر گرفتن قسمت دوم آن کمینه می شود. علاوه بر آن ما در این قسمت به جای پیدا کردن ماتریس اتصال، مسیرهای عبور ترافیک را محاسبه می کنیم که در آن توپولوژی شبکه و نحوه مسیریابی نهفته است. به عبارت دیگر الگوریتم مسیریابی ذکر شده در قسمت قبل نیز در حوزه مسأله بهینه سازی مشخص می شود. پاسخ مسأله در نهایت مسیر مناسب برای انتقال ترافیک میان هر دو زوج نقطه را در شبکه تعیین می کند به طوری که تأخیر کلی در سیستم کمینه شود.

مسأله بهینه سازی تأخیر را در یک شبکه می توان به صورت یک مسأله سریعترین انتقال^۱ [64]

[60] تعریف نمود. مسأله سریعترین انتقال بر روی یک شبکه دینامیک که شامل تعدادی نقطه ورودی و خروجی ترافیک است، تعریف می شود. هر ورودی میزان مشخصی ترافیک را وارد شبکه می کند که از

^۱ Quickest Transshipment Problem

خروجی متناظر آن خارج می شود. به این حجم ترافیک در اصطلاح تقاضای^۱ میان این ورودی و خروجی گفته می شود. یک شبکه دینامیک گرافی با ظرفیت های معین بر روی لبه های آن است. مسأله سریعترین انتقال پیدا کردن مسیرهایی بر روی گراف است که حجم معینی از مرسوله های تعریف شده بین ورودی ها و خروجی ها را در کمترین زمان منتقل نماید.

پاسخ به مسأله سریعترین انتقال در حوزه تئوری جریان شبکه مورد بحث قرار گرفته است [65] که در این حوزه به آن مسأله مینیمم هزینه جریان چند مرسوله گفته می شود.^۲ پاسخ به این مسأله به شکل متمرکز با داشتن اطلاعات تقاضاها و ظرفیت های شبکه، قابل محاسبه می باشد. ولی پاسخ های متمرکز که نیاز به داشتن اطلاعات کلی شبکه دارند، در کاربردهای عملی و در شبکه های بزرگ نیاز به تبادل حجم زیادی از اطلاعات کنترلی دارد و استفاده از آنها چندان عملی به نظر نمی رسد. اما با این وجود پاسخ به این مسأله به صورت متمرکز می تواند از جنبه های مختلف مفید واقع شود. به طور مثال این پاسخ می تواند به عنوان مرجعی برای مقایسه کارایی الگوریتم های پیشنهادی در فصول بعد مورد استفاده قرار گیرد. نکته قابل توجه آن است که همه الگوریتم های پیشنهادی نیاز به یک مرجع برای مقایسه دارند. با توجه به اینکه نتایج ارایه شده در بسیاری از تحقیقات مرتبط در شرایطی ارایه شده اند که برخی جزئیات الگوریتم و شرایط شبیه سازی به طور کامل مشخص نشده اند، انجام یک مقایسه عادلانه میان این روش ها و روش پیشنهادی امکان پذیر نمی باشد. پاسخ بهینه در این شرایط مرجع مناسبی برای مقایسه کارایی الگوریتم های پیشنهادی است. قبل از پرداختن به جزئیات این پاسخ تعاریف و اصول حل مسأله را مطرح خواهیم کرد.

^۱ Demand

^۲ Minimum-cost Multicommodity Flow Problem.

۱.۵.۳. تعاریف

- شبکه جریان^۱

در تئوری گراف یک شبکه جریان $G = (V, E)$ یک گراف جهت دار از نقاط به هم پیوسته است که در آن به هر لبه گراف ظرفیتی اختصاص داده می شود که جریان عبوری از آن نمی تواند از این ظرفیت بیشتر باشد.

- مسیر

یک مسیر P دنباله ای از نقاط $(n_1, n_2, \dots, n_k), k \geq 2$ است که شامل $k - 1$ لبه است. i امین لبه این مجموعه نقاط (n_i, n_{i+1}) و یا (n_{i+1}, n_i) را به هم متصل می کند.

- مرسوله

مرسوله i با یک مبدا و مقصد (s_i, t_i) و تقاضای ارسال میان آنها d_i تعریف می شود و به صورت (s_i, t_i, d_i) تعریف می شود. برای هر مرسوله تنها یک مبدا و یک مقصد تعریف می شود. ولی در شرایطی که چند مبدا و مقصد وجود داشته باشد تعریف بالا هنوز هم قابل استفاده است. در این شرایط تمامی منابع و یا مقاصد می توانند به یک منبع یا مقصد مجازی متصل شوند.

- جریان چند مرسوله

یک جریان چند مرسوله f بر روی گراف G مقدار جریان نامنفی $f^i(wv)$ را به هر لبه (wv) و مرسوله i تخصیص می دهد به طوری که f^i یک جریان تک مرسوله باشد. یک جریان چند مرسوله در شرایط بقاء^۲ که در روابط $(۸-۳)$ ، $(۹-۳)$ و $(۱۰-۳)$ تعریف شده است، صدق می کند. همچنین رابطه

^۱ Flow Network

^۲ Conservation

(۱۱-۳) شرط امکان پذیری^۱ این جریان را مشخص می کند.

$$\sum_{i, wv: wv \in E} f^i(wv) - \sum_{i, vw: vw \in E} f^i(vw) = 0 \quad v \neq s_i, t_i \quad (۸-۳)$$

$$\sum_{i, v: sv \in E} f^i(sv) - \sum_{i, v: vs \in E} f^i(vs) = d_i \quad (۹-۳)$$

$$\sum_{i, v: tv \in E} f^i(tv) - \sum_{i, v: vt \in E} f^i(vt) = -d_i \quad (۱۰-۳)$$

$$f(wv) = \sum_i f^i(wv) \leq u(wv) \quad wv \in E \quad (۱۱-۳)$$

• هزینه جریان^۲

هزینه یک جریان چند مرسوله f روی گراف جهت دار $G(V, E)$ در رابطه زیر تعریف می شود که $c^i(wv)$ هزینه عبور واحد جریان f^i از لبه wv است.

$$\sum_{i, wv: wv \in E} |c^i(wv)f^i(wv)| = CF \quad (۱۲-۳)$$

• مساله جریان شبکه می نیمم هزینه با چند مرسوله

هدف در یک مسأله جریان شبکه مینیمم هزینه با چند مرسوله، ارسال همزمان تعدادی مرسوله از طریق یک شبکه است به طوری که جریان کلی محدودیت ظرفیت لبه ها را رعایت کند و کل هزینه آن می نیمم باشد. خروجی این مسأله پس از بهینه سازی مقادیر $f^i(wv)$ خواهد بود. به عبارت دیگر مسأله بهینه سازی مشخص می کند که از هر پیوند چه جریانی باید عبور کند تا تأخیر کلی کمینه شود. بدیهی است که پیوندهایی که جریان آنها صفر است در شبکه overlay نهایی حذف می شوند. بنابراین پاسخ این مسأله بهینه سازی به صورت غیر مستقیم توپولوژی شبکه را نیز مشخص می کند. تفاوت

^۱ Feasibility

^۲ Flow Cost

پاسخ بهینه سازی حاصل از رابطه (۱۲-۳) با بهینه سازی رابطه (۷-۳) آن است که در رابطه (۷-۳) هزینه پیام های مبادله شده برای کنترل توپولوژی در نظر گرفته شده است. بنابراین پاسخ ناشی از این رابطه به شرایط واقع نزدیک تر و از کران می نیمم فاصله بیشتری دارد. در اینجا ما پاسخ بهینه ناشی از رابطه (۱۲-۳) را به عنوان کران می نیمم مسأله به دست خواهیم آورد.

۲.۵.۳. مدل سازی مسأله

ارسال پیام های به روز رسانی با کمترین تأخیر می تواند به صورت یک مسأله جریان شبکه می نیمم هزینه با چند مرسوله تعریف شود. در این مدل سازی شبکه زیرین که همان شبکه اینترنت است به عنوان شبکه جریان $G = (V, E)$ در نظر گرفته می شود. شبکه اینترنت و در نتیجه گراف G را می توان به صورت مش کامل فرض کرد. این فرض از آنجا ناشی می شود که در شبکه اینترنت هر نقطه می تواند با دانستن آدرس IP نقطه دیگر به صورت مستقیم از طریق سرویس مسیریابی لایه ۳ با آن ارتباط برقرار کند. بنابراین در شبکه اینترنت که در آن هر دو نقطه امکان ارتباط مستقیم را دارند فرض کامل بودن مش برقرار می باشد.

هر نقطه $v \in V$ در گراف G متناظر با اتواری در محیط مجازی است که تعدادی پیام به روز رسانی تولید می کند و پیام های به روز رسانی سایرین را که به خودش تعلق دارد یا باید عبور دهد را دریافت می کند. جریان های بین نقاط توسط لبه های شبکه جابجا می شود و هر لبه $e \in E$ نیز در گراف G متناظر با مسیری در لایه ۳ بر روی اینترنت است. ظرفیت هر لبه شبکه مانند $u(e)$ ، نامگذاری می شود و معادل پهنای باند در دسترس میان نقاط دو سر آن است. هزینه عبور واحد جریان از هر لبه، $c(e)$ ، برابر با تأخیر انتها به انتهای آن لبه است.

تعریف تقاضاها در شبکه به صورت کلی با مکانیزم مدیریت ناحیه مورد علاقه میان کاربران

ارتباط دارد. همانطور که قبلاً هم اشاره کردیم معمولاً هر نقطه با سایر نقاط ناحیه مورد علاقه اش پیام های به روز رسانی مبادله می کند و احتمال تبادل پیام ها با نقاط خارج ناحیه مورد علاقه یک نقطه کمتر است. از آنجایی که میزان علاقه نقاط به دریافت پیام های به روز رسانی معمولاً با فاصله از آنها نسبت دارد، می توان فرض کرد که تقاضای ارسال ترافیک میان هر منبع و مقصد به فاصله اتارهای متناظر آنها در محیط مجازی نسبت دارد. همچنین این تقاضا در شرایطی که فاصله مبدأ و مقصد از حد معینی بیشتر شود به صفر می رسد.

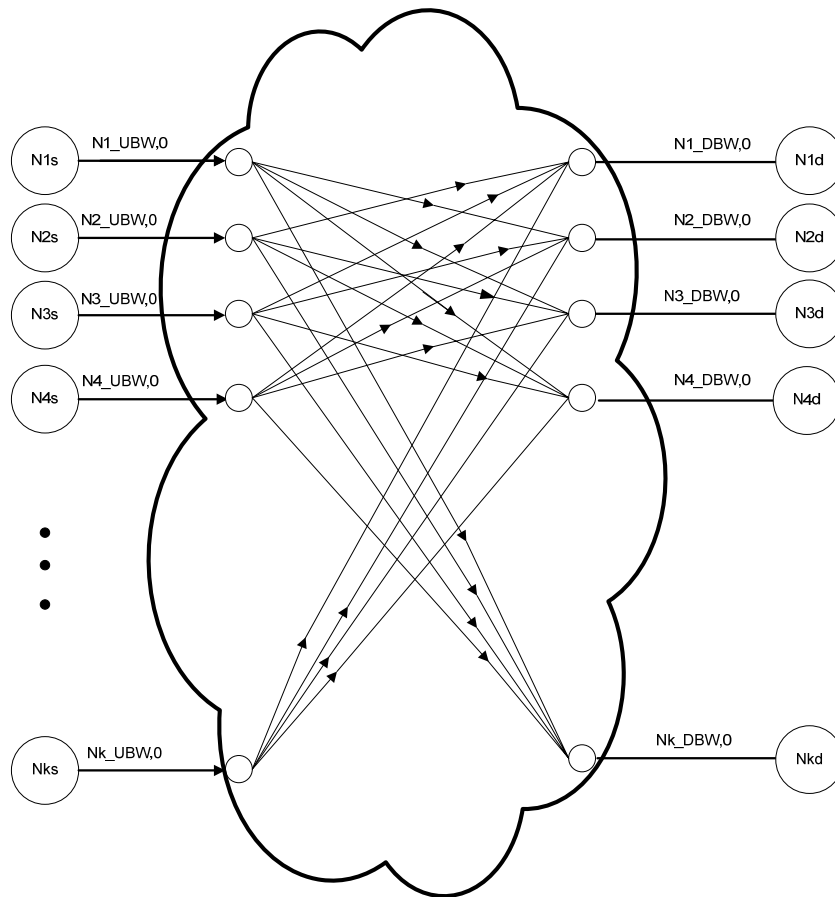
هدف ما در این مسأله انتخاب مسیرهایی است که به حداقل تأخیر منجر می شوند. همچنین در این انتخاب باید اولویت به مسیرهایی داده شود که حجم بیشتری از ترافیک را منتقل می کنند. بنابراین ما برای لبه ها هزینه ای متناسب با تأخیر آنها تعریف می کنیم و مسأله کنترل جریان را در شبکه حل خواهیم نمود.

شکل ۳-۳ مدل جریان شبکه را برای شبکه ای با K نقطه نشان می دهد. هر کاربر در محیط مجازی مسئول ایجاد و ارسال پیام های به روز رسانی مربوط به خود است. دیاگرام با نقاط که مسئول ایجاد پیام های به روز رسانی هستند شروع می شود.

نقاط شروع با پیوندهایی به اینترنت که به صورت یک ابر نشان داده شده است متصل می شوند. ظرفیت این پیوندها برابر پهنای باند ارسال^۱ این نقاط است. همچنین هزینه این پیوندها صفر در نظر گرفته می شود. علت این انتخاب آن است که تأخیر ثابت این قسمت در الگوریتم انتخاب مسیر تأثیری ندارد. در مدلسازی ارایه شده شبکه اینترنت با یک مش کامل مدل شده است که پیوندهای آن دارای ظرفیت نامتناهی و هزینه ای متناسب با تأخیر آنها دارد. فرض ظرفیت نامتناهی از آنجا ناشی می شود که

^۱ Upload

ظرفیت پیوندهای هسته اینترنت در مقابل ظرفیت پیوندهای ارتباط دهنده کاربران به اینترنت بسیار بیشتر است و ظرفیت ارسال به پهنای باند سمت کاربران محدود می شود.



شکل ۳-۳ دیاگرام شبکه جریان چند مرسوله ای

از آنجا که تمام عضوهای محیط مجازی در مجموعه دریافت کننده ها و ارسال کننده های پیام های به روز رسانی قرار دارند، مجموعه نقاط مبدا با مجموعه نقاط مقصد یکسان است. ظرفیت اتصال اینترنت به نقاط مقصد معادل با پهنای باند دریافت این نقاط است و هزینه این پیوندها مشابه با حالت ورود به اینترنت، صفر در نظر گرفته می شود.

هر نقطه در یک شبکه نظیر به نظیر باید بتواند پیام های به روز رسانی سایر نقاط را نیز برای رسیدن به مقصد نهایی انتقال دهد. بنابراین هر نقطه N_{is} در مجموعه مقصد باید با پیوندی با ظرفیت

بینهایت و هزینه صفر به نقطه متناظرش در مجموعه مبدأ N_{id} متصل باشد. انتخاب هزینه صفر به دلیل صرف نظر از تأخیر انتقال در این نقاط قسمت است. این پیوندها برای حفظ سادگی در شکل نشان داده نشده اند.

برای تعریف کامل مسأله همچنین به تعریف مرسوله ها نیاز است. همانطور که قبلاً اشاره کردیم، تقاضای تبادل پیام های به روز رسانی میان دو نقطه به فاصله آنها در محیط مجازی نسبت مستقیم دارد. بدون از دست دادن کلیت شکل فرض می کنیم که در شکل ۳-۳ اندیس نقاط معیاری از نشان دادن فاصله آنها در محیط مجازی است. بنابراین مرسوله (N_{is}, N_{id}, d) بین هر دو نقطه با تقاضای d تعریف می شود که مقدار این تقاضا با معکوس $|i - j|$ نسبت دارد.

شکل ۳-۳ مدلی از یک مسأله جریان چند مرسوله را نشان می دهد که از حل آن مسیرها و همسایگی های بهینه به دست خواهند آمد. مسأله جریان چند مرسوله یک مسأله کلاسیک در حوزه تحقیق در عملیات است و یافتن پاسخ های عددی برای آن سال های متعددی چالش در این حوزه بوده است: روش های تجزیه در جهت هزینه^۱ محدودیت های پیچیده را از مسأله حذف می کند و به جای آن هزینه ای را به پاسخ اضافه می کند. روش های تجزیه بر اساس منابع^۲ به هر مرسوله قسمتی از ظرفیت هر پیوند را اختصاص می دهد و بعد مسأله را به صورت تک مرسوله برای هر کدام از مرسوله ها حل می کند. گروه دیگری از روش ها به نام روش های تقسیم بندی^۳ از برنامه ریزی خطی و الگوریتم Simplex برای حل مسأله استفاده می کنند [66].

به علت بزرگ بودن ابعاد و تعداد متغیرهای مسأله کنترل جریان در کاربرد این رساله، ما برای

^۱ Price-directive Decomposition Approach

^۲ Resource-Directive Decomposition

^۳ Partitioning Methods

حل این مسأله از ابزار [67] ILOG (CPLEX) 10.0 استفاده نموده ایم که یکی از پیشرفته ترین ابزارهای مورد استفاده در بهینه سازی و برنامه ریزی خطی است. CPLEX برای حل مسئل بهینه سازی از روش Simplex استفاده می کند و می تواند پاسخ های دقیقی را برای مسأله کنترل جریان چند منظوره محاسبه نماید. پاسخ های حاصل از حل مدل بالا در فصل های بعد به عنوان مرجع مقایسه نتایج شبیه سازی مورد استفاده قرار خواهند گرفت.

فصل چهارم

ساختار مبتنی بر DHT

مقدمه ای بر DHT

استفاده از DHT در محیط های مجازی

روش های کاهش تأخیر مسیریابی در

نسبت دادن ID های مناسب

نتایج شبیه سازی

۱.۴. مقدمه ای بر DHT

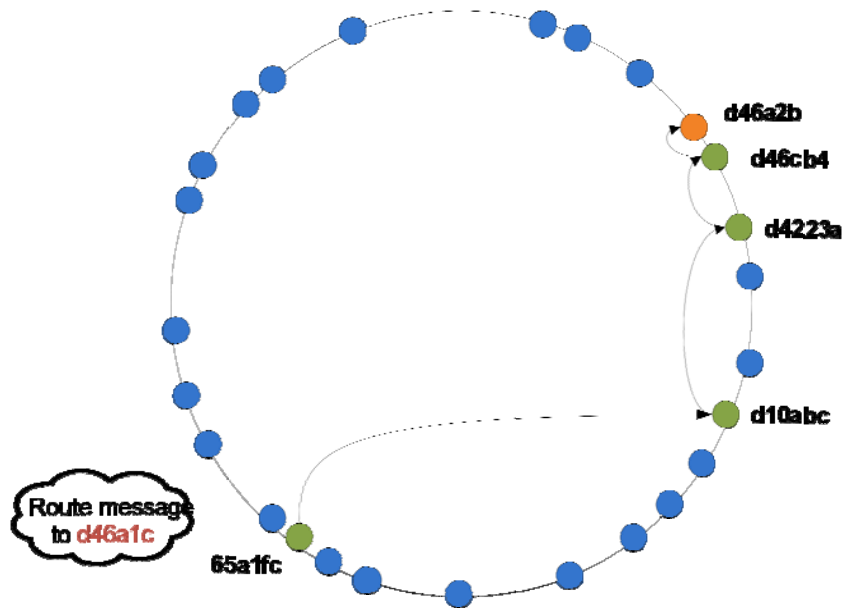
شبکه های DHT خانواده ای از سیستم های گسترده هستند که نقاط آنها می توانند اطلاعات مورد جستجوی خود را با داشتن کلید آنها به آسانی بازیابی نمایند. شبکه های نظیر به نظیر اشتراک فایل عموماً از DHT به عنوان روشی مناسب به منظور جستجو و ذخیره اطلاعات در یک سیستم گسترده با ابعاد بزرگ، استفاده نموده اند. CAN ، Chord ، Tapestry و Pastry از معروفترین شبکه های نظیر به نظیر مبتنی بر DHT هستند. این شبکه های ساختار یافته نظیر به نظیر از DHT به عنوان بستری برای ذخیره مکان اطلاعات استفاده می کنند که در آن هر قسمتی از اطلاعات در نقطه ای که متناظر با کلید محاسبه شده از روی آن است ذخیره می شود. بنابراین مکان ذخیره هر قسمتی از اطلاعات با داشتن کلید آن مشخص می شود و اطلاعات با در دست داشتن کلیدهای متناظر با آنها به صورت گسترده قابل ذخیره سازی و جستجو هستند.

در میان شبکه های فوق Tapestry قدیمیترین و پیچیده ترین آنهاست. سایر شبکه ها از جمله

CAN ، Chord و Pastry شباهت های بسیاری در اصول کارکردشان دارند . قسمتی از تفاوت این شبکه ها در فضای نگاشت شناسه^۱ نقاط آنها می باشد. CAN از یک دستگاه مختصات چند بعدی برای جای دادن شناسه نقاط استفاده می کند. Chord از یک فضای دایره ای تک جهت برای شناسه نقاطش بهره می گیرد، در حالی که Pastry و Tapastry از شبکه های مش Plaxton استفاده می کنند. در میان این شبکه ها Pastry آزادی بیشتری را در انتخاب همسایگی ها برای نقاط شبکه ایجاد می کند. بنابراین شبکه Pastry انعطاف پذیری بیشتری برای ایجاد یک ساختار جدید بر روی DHT فراهم می آورد. علاوه بر آن Scribe یک مکانیزم چند پخشی مناسب بر روی Pastry است که امکان چند پخشی مناسبی را بر روی شبکه Pastry ایجاد می کند.

به هر نقطه در شبکه Pastry یک شناسه با طول ۱۲۸ بیت اختصاص داده می شود و شناسه نقاط به صورت همگن در فضای شناسه ها توزیع می شود. مکانیزم مسیر یابی Pastry می تواند هر بسته با یک کلید مشخص را با عبور از تعداد محدودی نقطه میانی به سمت نقطه ای با نزدیکترین شناسه به کلید بسته هدایت نماید. در یک شبکه Pastry با N نقطه، هر بسته با عبور از حداکثر $\log_{2b}(N)$ به مقصد خود خواهد رسید که b یک پارامتر مسیریابی است که مقدار پیش فرض آن ۴ است. شکل ۴-۱ نمونه ای از مراحل مسیر یابی بسته ای به مقصد d46a2b را که مسیرش از نقطه 65a1fc شروع شده است، نشان می دهد. همانطور که در شکل دیده می شود در عبور از هر نقطه میانی، شناسه نقطه دریافت کننده نسبت به نقطه قبلی یک بیت مشترک بیشتر با مقصد بسته دارد.

^۱ ID(Identification)



شکل ۴-۱ نمونه مسیریابی یک بسته در Pastry

۲.۴. استفاده از DHT در محیط های مجازی

یکی از بزرگترین محاسن استفاده از DHT در کاربرد محیط های مجازی گسترده حذف سرپاره ناشی از ارسال تقاضا برای یافتن آدرس IP نقاط است. اما در عین حال شبکه های DHT در ساختار فعلی قادر به برآورده کردن برخی از نیازهای اساسی کاربردهای محیط های مجازی نمی باشند. در اینجا به برخی از این تفاوت ها اشاره خواهیم نمود.

- تقاضای تبادل پیام های به روز رسانی به صورت یکنواخت در میان تمام نقاط محیط مجازی توزیع نشده است و توزیع آن بسته به فاصله مجازی نقاط از یکدیگر تغییر می کند. این توزیع به علت وجود مکانیزم های کنترل ناحیه مورد علاقه در کاربران می باشد. همانطور که قبلاً اشاره کردیم، اوتارها با نقاط موجود در ناحیه مورد علاقه شان که بیشتر در اطرافشان قرار دارند، پیام های به روز رسانی مبادله می کنند. بنابراین توزیع ترافیک در میان نقاط یک محیط مجازی تابع فاصله نقاط از یکدیگر است و نقاط نزدیک به یکدیگر تمایل بیشتری برای آگاهی از

یکدیگر دارند. چنین توزیعی در کاربردهای اشتراک فایل که تقاضاها به صورت تصادفی و یکنواخت در شبکه توزیع شده اند، وجود ندارد.

- کاربران محیط مجازی به صورت دائم و با نرخ متغیری که تابع حالت تعامل اوتار متناظر با آنهاست پیام های به روز رسانی مبادله می کنند. اوتارهایی که در نواحی پر ازدحام محیط مجازی قرار می گیرند نقاط بیشتری را در اطراف خود دارند و نیاز به ارسال و دریافت حجم بیشتری از پیام های به روز رسانی دارند. علاوه بر آن برخی از اوتارها به دلیل حرکت بیشتر و سریعتر ممکن است نیاز بیشتری به تبادل پیام های به روز رسانی داشته باشند. چنین رفتاری در کاربردهای اشتراک فایل وجود ندارد.

- محیط های مجازی به منظور نمایش بلادرنگ اطلاعات نیاز به ارسال پیام های به روز رسانی با کمینه مقدار تأخیر دارند. از سوی دیگر مسیریابی DHT در کاربردهای شبکه های اشتراک فایل تنها برای جستجوی اطلاعات استفاده می شود و مسیر اصلی انتقال پس از یافتن اطلاعات مورد نظر به صورت مستقیم می باشد. بنابراین تأخیر مسیر یابی DHT در این شبکه ها مورد توجه چندانی نبوده است.

- محیط های مجازی گسترده در مقایسه با شبکه های اشتراک فایل ساختارهایی بسیار دینامیک و با تغییرات بیشتر دارند. این تغییرات به علت تغییر در مکان اوتارها و همچنین ورود و خروج نقاط به محیط مجازی ایجاد می شود. در چنین شرایطی به روز نگاه داشتن جدول های مسیر یابی و اطلاعات توپولوژی یک چالش بزرگ محسوب می شود.

- شبکه های DHT در تشکیل ساختار overlay از کنار هم قرار دادن نقاط نزدیک در حوزه جغرافیایی پرهیز می کنند تا در صورت ایجاد مشکل در یک ناحیه از شبکه اینترنت، قسمت به

هم پیوسته ای از overlay دچار مشکل نشود. ولی در مقابل در محیط های مجازی اتصال نقاط نزدیک در محیط مجازی در overlay مطلوب می باشد و باعث کاهش نقاط میانی در ارسال پیام های به روز رسانی میان نقاط نزدیک می شود.

- ساخت overlay برای محیط های مجازی به مفهوم برقراری ارتباطات با تأخیر کم میان نقاط با توزیع ارسال دسته بندی شده^۱ است. در مقابل DHT برای مسیر یابی ترافیک با کمترین تعداد گام های میانی و در حالت توزیع یکنواخت طراحی شده است.

بنابراین استفاده از DHT در کاربردهای محیط های مجازی نیاز به ایجاد تغییرات اساسی در طراحی آنها دارد تا مسایل مربوط به تفاوت های بالا حل شوند. اما در مقابل استفاده از DHT در کاربردهای محیط مجازی مزیت های متعددی مانند قابلیت توسعه ، تعادل بار شبکه و آدرس دهی گسترده اطلاعات را به همراه دارد. بنابراین در این قسمت هدف ما بر آن است که ساختاری را بر مبنای DHT برای محیط های مجازی پیشنهاد دهیم و با ایجاد تغییرات لازم ویژگی های نامطلوب DHT را برای بکار گیری در محیط های مجازی جبران سازی نماییم. راه حل ارایه شده به عنوان مکملی برای مسیر یابی DHT عمل می کند. در ساختار پیشنهادی، ایده هایی همچون تخصیص شناسه به نقاط و مسیر یابی سلسله مراتبی بر مبنای شناسه نقاط، مشابه شبکه های DHT تعریف می شود. ولی به نقاط شبکه DHT هوشمندی بیشتری در زمینه انتخاب همسایه های نزدیک و ارسال بسته ها از مسیرهای کوتاه، اضافه می شود.

ساختار پایه DHT مورد استفاده بر مبنای Pastry می باشد که در میان معروفترین شبکه های DHT قرار دارد. از سوی دیگر قابلیت انعطاف در ساختار آن بیشتر می باشد و بنابراین قرار دادن

^۱ Clustered

الگوریتم های مکمل در کنار آن ساده تر انجام می شود. در قسمت های بعدی این فصل راه حل های پیشنهادی در جهت جبران سازی تأخیر در شبکه های DHT با استفاده از ویژگی های خاص محیط های مجازی، ارایه خواهد شد.

۳.۴. روش های کاهش تأخیر مسیریابی در DHT

طراحی شبکه های DHT با تأخیر کم در سالهای اخیر موضوع دسته ای از تحقیقات بوده است. در جریان این تحقیقات، تکنیک های مختلفی برای کاهش تأخیر در شبکه های DHT پیشنهاد شده است [68]. در گروهی از این راه کارها، انتخاب همسایه های نزدیک به عنوان یک روش بسیار کارآمد در کاهش تأخیر معرفی شده است [71]-[69]. انتخاب مسیر های نزدیک هنگام ارسال بسته ها روش دیگری در کاهش تأخیر است که در تحقیقات دیگری مانند [72] پیشنهاد شده است. در این روش، تأخیر ارسال بسته از مسیرهای مختلف بر مبنای اطلاعات همسایگان تخمین زده می شود و بهترین نقطه بعدی برای ارسال بسته بر مبنای این تخمین انتخاب می شود. دسته دیگری از روش ها مانند [73] سعی در بهبود تدریجی تأخیر DHT در جریان مسیریابی دارند. این بهبود تدریجی با استفاده از اطلاعات موجود در نمونه های تصادفی انتخاب شده از میان درخواست های جستجو که در شبکه در حرکت هستند، انجام می شود. در [74] روش دیگری برای انتخاب همسایه ها بر مبنای اطلاعات آدرس IP پیشنهاد شده است.

روش پیشنهادی به منظور کاهش تأخیر در این رساله ، در دو مرحله اصلی انجام می شود. در مرحله نخست، شناسه های مناسب برای نقاط انتخاب می شوند و در مرحله دوم روش هایی جهت انتخاب همسایه های نزدیک برای جدول مسیریابی و انتخاب کوتاه ترین مسیر هنگام ارسال بسته ها ارایه خواهد شد. شبیه سازی ها نشان می دهند که به کار گیری این دو مرحله در کنار یکدیگر تأخیر

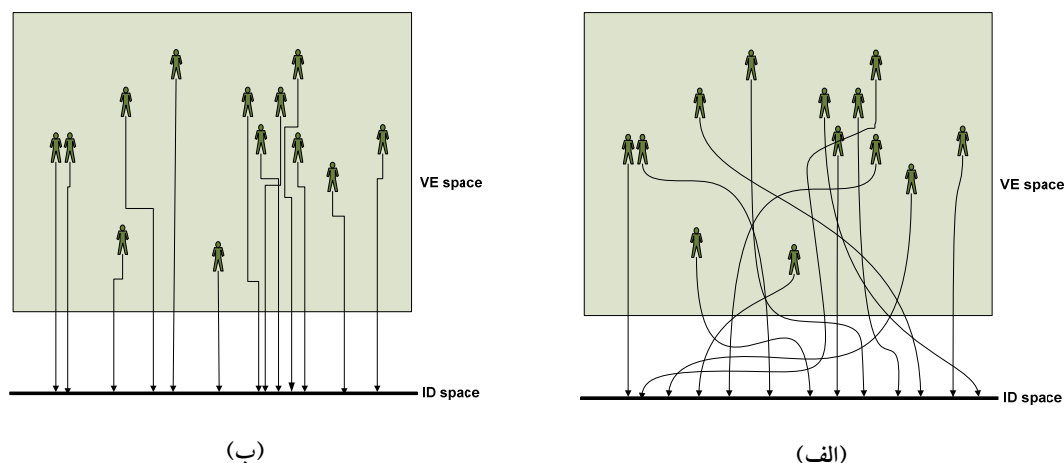
ناخواسته شبکه های DHT را تا حد قابل قبولی کاهش خواهد داد.

۱.۳.۴. نسبت دادن ID های مناسب

بیشتر شبکه های DHT نقاط شبکه خود را به شناسه های تصادفی می نگارند تا هنگام بروز مشکل در یک ناحیه از شبکه زیرین این اشکال منجر به از کار افتادن قسمتی پیوسته از شبکه overlay نشود. نگاشت تصادفی شناسه ها برای کاربردهایی مانند شبکه های اشتراک فایل که توزیع ترافیک در آنها یکنواخت است، مسأله خاصی به وجود نمی آورد. در مقابل در کاربرد محیط های مجازی گسترده توزیع ترافیک تصادفی نبوده و به صورت دسته بندی شده می باشد. بنابراین هر نقطه تمایل یکسان برای تبادل پیام با تمام نقاط ندارد و تقاضای تبادل پیام ها با فاصله نقاط نسبت معکوس دارد. از آنجا که مسیر یابی مورد استفاده مشابه با مسیریابی شبکه Pastry بوده که در آن معیار تعداد گام ها، فاصله عددی میان شناسه هاست، بهتر است برای کاهش تعداد گام های میانی در عبور ترافیک، نقاط نزدیک به هم به شناسه های نزدیک نگاشته شوند. بدین ترتیب پیام های به روز رسانی مربوط به یک اوتار که باید در همسایگی آن توزیع شوند برای رسیدن به مقصد ناچار به عبور از نقاط دوردست در شبکه نیستند. در نتیجه کارایی الگوریتم مسیر یابی بهبود یافته و تأخیر آن کاهش می یابد.

برای استفاده از شبکه های Pastry که از فضای شناسه یک بعدی استفاده می کنند، هر نقطه باید یک شناسه تک بعدی داشته باشد. با توجه به توضیحات قبلی، شناسه های نقاط را به گونه ای انتخاب می کنیم که نقاط در یک همسایگی شناسه های نزدیک به هم داشته باشند. به این نگاشت در اصطلاح، نگاشت مبتنی بر مکان^۱ گفته می شود. شکل ۲-۴ نگاشت مبتنی بر مکان را در مقابل نگاشت تصادفی نشان می دهد.

^۱ Location-Based Mapping



شکل ۲-۴ تخصیص شناسه (الف به شکل تصادفی (ب مبتنی بر مکان

به منظور انجام نگاشت مبتنی بر مکان بر روی نقاط محیط مجازی به یک تابع مناسب نگاشت نیاز است که نزدیکی نقاط را به بهترین صورت در هنگام نگاشت حفظ نماید. بدین منظور در این رساله استفاده از منحنی های پرکننده فضا^۱ پیشنهاد می شود که هنگام نگاشت نقاط از فضای چند بعدی به فضای تک بعدی نزدیکی آنها را به بهترین شکل حفظ می کنند. بنابراین منحنی پرکننده فضا در واقع جایگزین تابع hash مورد استفاده در شبکه های DHT می شود.

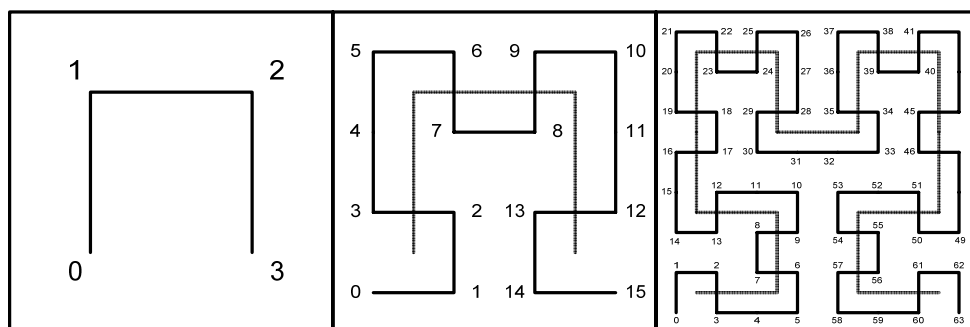
نمونه های متعددی از منحنی های پرکننده فضا در حوزه ریاضیات پیشنهاد شده اند. منحنی های پرکننده فضا می توانند نگاشت میان نقاط یک فضای چند بعدی را به فضای تک بعدی انجام دهند. این منحنی ها معمولاً از یک نقطه شروع و با یک الگوریتم خاص گسترش یافته و تمامی نقاط فضا را پوشش می دهند. منحنی ترتیب Z [75]، کد Gray [76] و منحنی هیلبرت [77] از معروف ترین منحنی های پرکننده فضا به شمار می روند که در میان آنها منحنی هیلبرت به عنوان بهترین نگاشت از لحاظ حفظ نزدیکی اثبات شده است [79][78]. بنابراین ما از منحنی هیلبرت برای نگاشت مکان نقاط به شناسه های آنها استفاده کرده ایم.

^۱ Space-Filling Curve

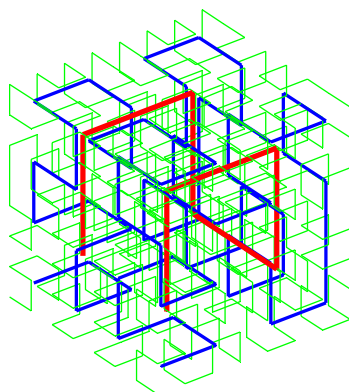
منحنی های هیلبرت fractal های متقارن و پیوسته ای هستند که از واحدهای اساسی که در اصطلاح فنجان گفته می شود، تشکیل شده اند. منحنی هیلبرت مرتبه اول از یک تک فنجان تشکیل شده است و مرتبه های بالاتر منحنی هیلبرت با انجام تبدیل هایی مانند چرخش و انتقال از منحنی های مرتبه های پایین تر به دست می آید. یک منحنی هیلبرت مرتبه اول در فضای دو بعدی می تواند یک صفحه 2×2 را پر کند. منحنی مرتبه دوم، این فنجان پایه را با چهار فنجان کوچکتر جایگزین می کند که با تعدادی مفصل به هم متصل شده اند و این فرآیند بازگشتی جایگزینی هر فنجان با چهار فنجان کوچکتر و سه مفصل برای ساخت منحنی های مرتبه بالاتر تکرار می شود.

یک منحنی هیلبرت مرتبه N فضایی D بعدی را با 2^{ND} رأس می پوشاند به طوری که منحنی از هر رأس تنها یک بار می گذرد و هر پیوند آن دو نقطه نزدیک به هم را در فضا متصل می کنند. شکل ۳-۴ فرآیند بازگشتی تشکیل منحنی هیلبرت را در فضای ۲ بعدی برای مرتبه های اول تا سوم نشان می دهد.

مرتبه منحنی هیلبرت چگالی نقاط روی منحنی را مشخص می کند. بنابراین، یک منحنی از مرتبه بالا می تواند تعداد نقاط زیادی را در صفحه بپوشاند و برای نگاشت نقاط در یک محیط مجازی با کاربران زیاد مناسب است. اگر چگالی نقاط منحنی هیلبرت به میزان کافی نباشد، این امکان وجود دارد



شکل ۳-۴ منحنی هیلبرت مرتبه ۱ تا ۳

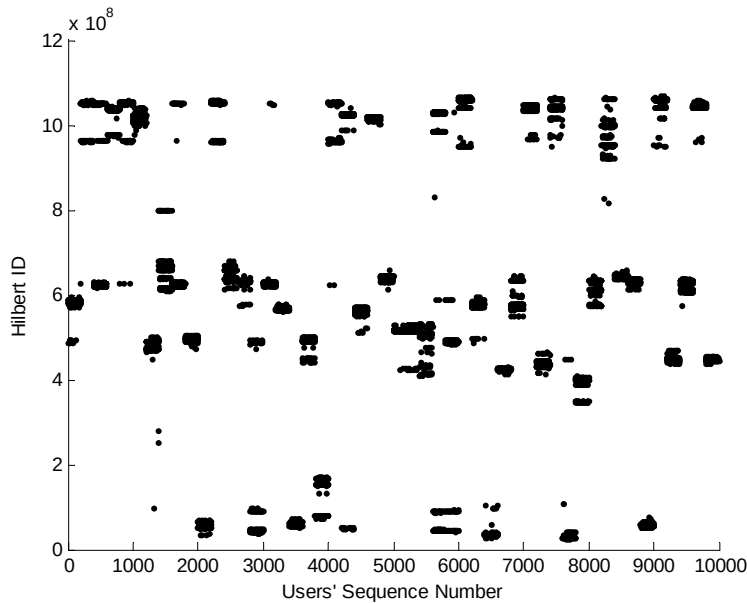


شکل ۴-۴ منحنی هیلبرت سه بعدی مرتبه ۳-۱

که دو یا چند مکان در محیط مجازی به یک نقطه روی منحنی هیلبرت نگاشته شوند و در اصطلاح تصادم اتفاق بیفتد. از سوی دیگر اگر مرتبه منحنی هیلبرت خیلی زیاد و بیشتر از حد نیاز باشد، هزینه های نگاشت نقاط و نگهداری جدول های مسیر افزایش می یابد. بنابراین مرتبه منحنی هیلبرت باید بر مبنای مصالحه میان این پارامترها تعیین شود.

برای انتخاب مرتبه مناسب منحنی هیلبرت می توان چگالی نقاط روی منحنی را به حدکافی زیاد در نظر گرفت به گونه ای که هیچ اوتاری میان دو نقطه جای نگیرد. بدین ترتیب در نگاشت اوتارها به نقاط منحنی تصادمی رخ نخواهد داد. نکته قابل توجه این است که در شبیه سازی ها ما اوتارها را نقاط با ابعاد صفر در نظر گرفته ایم. بنابراین انتخاب مرتبه مناسب برای منحنی هیلبرت در شبیه سازی ها به این روش نمی تواند انجام شود. به این منظور ما مقدار تصادفی اولیه ای را برای مرتبه منحنی در نظر گرفته و نگاشت را انجام دادیم تا تعداد تصادم ها مشخص شود، پس از آن با آزمایش و خطا مرتبه را تا حدی زیاد نمودیم که تعداد تصادم ها بسیار کم باشد.

برای نمایش ویژگی های حفظ همسایگی در نگاشت هیلبرت، منحنی مرتبه ۱۰ هیلبرت برای نگاشت توزیع نشان داده شده در شکل ۲-۳ به کار گرفته شده است. نتایج این نگاشت در شکل ۴-۵



شکل ۴-۵ نگاشت مکان کاربران به وسیله منحنی هیلبرت

نشان داده شده است.

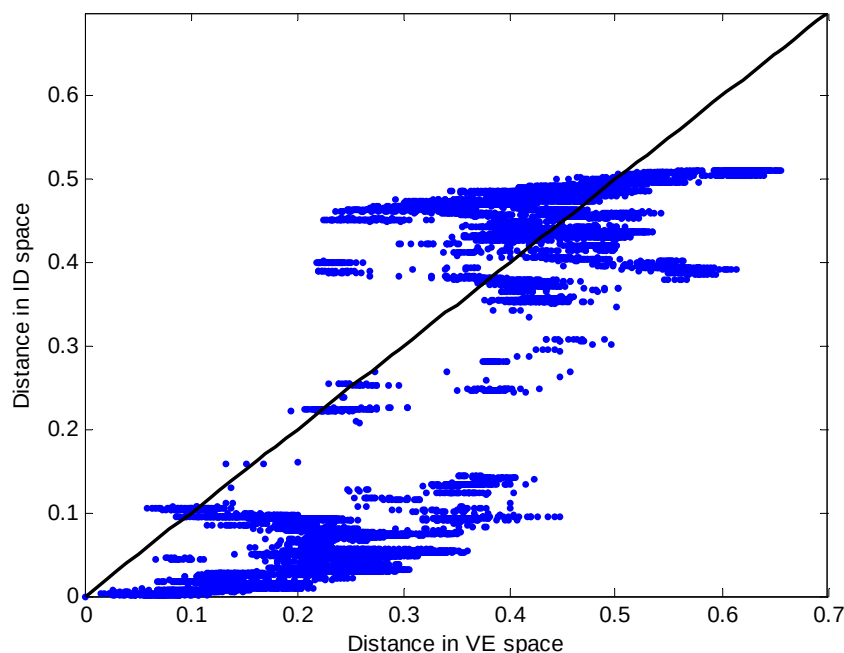
همان طور که در شکل دیده می شود، توزیع دسته بندی شده نقاط در محیط مجازی در فضای شناسه ها هم وجود دارد. علت توزیع غیر یکنواخت نقاط در فضای شناسه ها، یکنواخت نبودن توزیع کاربران نسبت به محور Z است. این مسأله سبب می شود که برخی نواحی که متناظر با مقادیر Z بزرگ هستند، پس از نگاشت خالی بمانند.

اگر محیط مجازی به شکل یک مکعب با مختصات های کمینه و بیشینه $(0,0,0)$ تا $(x_{\max}, y_{\max}, z_{\max})$ در نظر گرفته شود، فاصله نرمالیزه شده دو نقطه v_i و v_j از رابطه (۴-۱) قابل محاسبه می باشد:

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} / \sqrt{(x_{\max})^2 + (y_{\max})^2 + (z_{\max})^2} \quad (۴-۱)$$

به صورت مشابه فاصله نرمالیزه دو نقطه در فضای شناسه ها می تواند به صورت زیر تعریف

شود:



شکل ۴-۶ فاصله نرمالیزه در فضای محیط مجازی و فضای شناسه ها

$$ID_i - ID_j / ID_{\max} \quad (۲-۴)$$

برای حفظ فاصله نقاط در نگاشت، فاصله نرمالیزه در فضای محیط مجازی و فضای شناسه ها باید تا حد امکان به هم نزدیک باشند و در حالت ایده آل که نگاشت قادر است نسبت فاصله ها را به طور کامل حفظ کند، فاصله نرمالیزه در این دو فضا باید مساوی باشد. بنابراین حالت ایده آل با خط $x=y$ نشان داده می شود. البته قابل توجه است که حالت ایده آل به علت کاهش ابعاد فضا از ۳ به ۱ هنگام نگاشت، قابل رخ دادن نمی باشد. بنابراین وجود مقداری انحراف در فاصله نسبی طبیعی به نظر می رسد. شکل ۴-۶ فاصله نرمالیزه شده را در هر دو محیط در کنار خط $x=y$ به عنوان حالت ایده آل نشان می دهد. همانطور که در شکل دیده می شود، نقاط تا حد قابل قبولی به خط $x=y$ نزدیک هستند.

به منظور نمایش بهتر ویژگی های نگاشت مورد استفاده، در اینجا پارامتر دیگری را به نام

انحراف فاصله به صورت زیر تعریف می کنیم:

$$Distance\ deviation\ (M,N)= \left| \frac{\sqrt{(M_x - N_x)^2 + (M_y - N_y)^2 + (M_z - N_z)^2}}{\sqrt{X_{max}^2 + Y_{max}^2 + Z_{max}^2}} - \frac{|ID_M - ID_N|}{ID_{max}} \right| \quad (3-4)$$

که در آن X_{max} ، Y_{max} و Z_{max} بزرگترین مختصات در محیط مجازی و ID_{max} بزرگترین شناسه

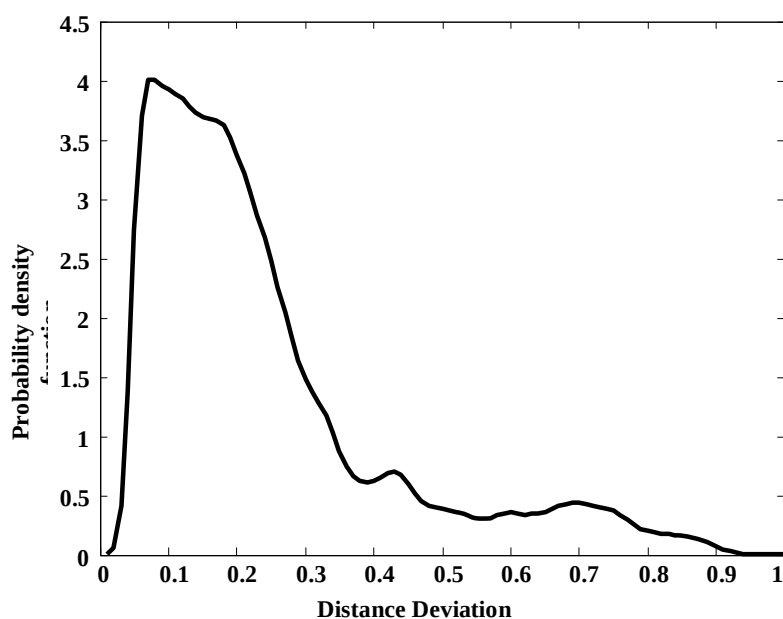
ممکن روی منحنی هیلبرت می باشد.

شکل ۷-۴ تابع توزیع انحراف فاصله را برای نقاط محیط مجازی نشان می دهد. اگر نقاط به

صورت تصادفی در فضای شناسه ها نگاشته شده بودند، این توزیع به شکل یکنواخت در بازه [0 1] می

شد. ولی در شکل دیده می شود که این توزیع بیشتر در اطراف مقادیر کوچکتر متمرکز است. بنابراین

نگاشت هیلبرت انحراف فاصله را تا حد قابل قبولی کاهش داده است.



شکل ۷-۴ توزیع انحراف فاصله

۲.۳.۴. انتخاب مسیرها با توجه به تأخیر

انتخاب مناسب شناسه ها اولین قدم برای ساخت یک شبکه overlay مناسب است. با این وجود کماکان مسیرهای متعددی برای عبور ترافیک در شبکه وجود دارند که از نظر تأخیر مسیر با هم متفاوت هستند. بنابراین به منظور کمینه کردن تأخیر، فاصله فیزیکی نقاط بر روی شبکه اینترنت باید در انتخاب مسیرها منظور شود تا فاصله نقاط میانی مسیر از نظر تأخیر کم باشد.

انتخاب مسیرهای کوتاه از لحاظ تأخیر در دو مرحله صورت می پذیرد: مرحله نخست در انتخاب عناصر جدول مسیریابی به اجرا درمی آید و مرحله دوم در انتخاب بهترین همسایه هنگام ارسال بسته ها تأثیر می گذارد. الگوریتم پیشنهادی بهبود تأخیر در این رساله هر دو سیاست های مناسبی را در هر دو مرحله اجرا می کند تا بهترین نتایج در کاهش تأخیر حاصل شود.

در ساختار اصلی Pastry روشی برای انتخاب همسایه های نزدیک^۱ (PNS) پیشنهاد شده است. در این روش هر نقطه سعی می کند نقاط نزدیک به خود که تأخیر ارسال کمتری با آنها دارد، را پیدا کند. پیدا کردن همسایه های نزدیک در حالت بهینه نیاز به تبادل $O(N)$ پیام در یک شبکه N نقطه ای دارد و بنابراین هزینه زیادی را به دنبال خواهد داشت. بنابراین Pastry از یک روش ابداعی برای کاهش این هزینه به $O(\log(N))$ استفاده می کند. اما نکته قابل توجه آن است که انتخاب همسایه های نزدیک در Pastry تنها هنگام پیوستن نقاط به شبکه انجام می شود. بنابراین این روش در کاربرد محیط های مجازی به دلیل ویژگی های دینامیک آن که نقاط به طور مداوم مکان و شناسه خود را تغییر می دهند، قابل پیاده سازی نمی باشد. [71] روشی ابتدایی را برای محلی سازی ارتباطات در Pastry پیشنهاد داده است که در تحقیقات بعدی بهبود یافته است. نگاه داری همسایگان متعدد در هر خانه جدول مسیر یابی

^۱ Proximity Neighbor Selection

و انتخاب یک کاندیدای مناسب از میان آنها بر مبنای مقصد بسته، مفهوم مشترک میان بسیاری از این روش ها است. هر خانه جدول مسیر یابی باید نقطه ای را که شناسه آن در محدوده خاصی قرار دارد ذخیره کند. بنابراین هر نقطه می تواند هنگام انتقال یک بسته از میان نقاط که شناسه آنها پسوند مناسب را دارند، نقطه را که از نظر فیزیکی به آن نزدیک تر است را برای انتقال بسته انتخاب کند. چگونگی این انتخاب اصل متمایز کننده الگوریتم های مختلف انتخاب همسایه های نزدیک است.

نکته قابل توجه در مورد تحقیقات مرتبط در بهینه سازی تأخیر شبکه های DHT آن است که بیشتر این ساختارها کارایی مناسب را در کاربرد محیط های مجازی گسترده و در حضور شرایط دینامیک، دارا نمی باشند. در این رساله روش گسترده ای به منظور انتخاب همسایه های نزدیک در محیط های مجازی ارائه می شود که قادر به حفظ کارایی در حضور تغییرات در شبکه است. بیشتر روش های پیشین ارائه شده برای انتخاب مسیر های کوتاه در ساختارهای گسترده معمولاً به روش حریص^۱ عمل می کنند. این بدان معنا است که در هر مرحله تنها فاصله با نقطه بعدی انتقال بسته کمینه می شود و این لزوماً به معنای کمینه شدن فاصله در کل مسیر نمی باشد. با وجود آنکه روش های حریص قادر به یافتن کوتاهترین مسیرها نیستند، پاسخ آنها با توجه به نیاز به حوزه محدود اطلاعات در هر نقطه در مسایل بسیاری کارایی مناسب را دارا است.

نکته قابل توجه آن است که درصد بهینه بودن پاسخ این مسأله به میزان اطلاعات موجود در هر نقطه بستگی دارد و این در حالی است که جمع آوری و به روز رسانی این اطلاعات نیاز به صرف منابع پردازشی و انتقالی در شبکه دارد. پاسخ های مناسب برای مسأله از مصالحه میان این پارامترها حاصل می شوند. در ادامه ما روش بهینه سازی تأخیر را بر مبنای بهینه سازی با گروه مورچه ها پیشنهاد خواهیم

^۱ Greedy

کرد.

۳.۳.۴. جستجوی مسیرهای مناسب با روش گروه مورچه ها

جستجوی همسایگی های نزدیک در یک شبکه overlay به مسأله طراحی حرکت دینامیک^۱ در شبکه شباهت بسیاری دارد که در آن الگوریتم طراحی مسیر باید در مقابل تغییرات لحظه ای در شبکه پاسخ های سریع از خود نشان دهد.

در حوزه بهینه سازی الگوریتم های قاطع در اصطلاح به الگوریتم هایی گفته می شود که در مقابل پاسخ های یکسان همیشه خروجی های یکسانی را نتیجه می دهند. این پاسخ معمولاً مینیمم یا ماکزیمم عمومی تابع هدف مسأله است. اما در بسیاری از کاربردها به دست آوردن یک پاسخ تقریبی در زمان کمتر و با اطلاعات کمتر ارزشمندتر از به دست آوردن پاسخ بهینه است. روش های تقریبی بهینه سازی متعددی برای این منظور پیشنهاد شده اند که بهینه سازی با روش گروه مورچه ها^۲ یکی از این روش ها می باشد.

روش بهینه سازی با گروه مورچه ها (ACO) که بر مبنای رفتار واقعی گروه مورچه ها در طبیعت پایه ریزی شده است، به عنوان یک روش ابداعی برای حل مسایل ترکیبی بهینه سازی به کار می رود [80]. ACO یک الگوریتم تکاملی محسوب می شود که در آن گروهی از نقاط هوشمند با همکاری یکدیگر و با یک روش آماری سعی در به دست آوردن پاسخ بهینه یک مسأله بدون نیاز به داشتن اطلاعات متمرکز مسأله در یک نقطه دارند. در طبیعت، وسیله برقراری ارتباط میان مورچه ها ماده ای به نام فرومون است که مورچه ها هنگام راه رفتن از خود به عنوان ردپا به جای می گذارند. هر مورچه با

^۱ Dynamic Motion Planning

^۲ Ant Colony Optimization

حس کردن فرومون ها قادر به تشخیص مسیر حرکت سایر مورچه ها است. بنابراین فرومون ها به مورچه ها کمک می کنند تا به جای حرکت در مسیرهای تصادفی و جستجو به دنبال غذا، مسیر هایی مشابه سایر مورچه ها را دنبال کنند. چنین مکانیزم ساده ای به شکل تدریجی به مورچه ها امکان می دهد تا کوتاه ترین مسیر را به سمت غذا پیدا کنند. به طور مثال اگر دو مسیر به سمت غذا وجود داشته باشد که یکی از دیگری طولانی تر باشد، در ابتدا مورچه ها در هر دو مسیر حرکت می کنند. با گذر زمان مورچه هایی که مسیر کوتاه تر را پیموده اند، در زمان کمتری غذا ها را به لانه می برند و دوباره به محل غذا باز می گردند. بنابراین مسیر لانه تا غذا را تعداد بار بیشتری می پیمایند و فرومون های بیشتری در این مسیر می ریزند. در نتیجه مورچه های بیشتری این مسیر را انتخاب می کنند.

به طور مشابه در الگوریتم ACO هر نقطه سعی می کند بر مبنای فرومون سایر نقاط پاسخی را برای مسأله محاسبه کند و بر مبنای کیفیت پاسخ محاسبه شده بر روی آن مقداری فرومون بریزد. نکته اساسی در طراحی یک الگوریتم ACO مدیریت چگونگی و میزان ریختن فرومون بر روی پاسخ ها است.

در این رساله ما از روش ACO برای یافتن پاسخ بهینه به مسأله جستجوی نزدیکترین همسایگان استفاده می کنیم. نکته قابل توجه این است که روش پیشنهادی نیازی به جمع آوری متمرکز اطلاعات توپولوژی در یک نقطه ندارد و نقاط با داشتن اطلاعات همسایگان خود در یک الگوریتم بهینه سازی گسترده شرکت می کنند. علاوه بر آن مورچه های شرکت کننده در الگوریتم بهینه سازی به صورت مستقل در شبکه حرکت نمی کنند و بر روی پیام های به روز رسانی سوار می شوند.¹ نخستین حسن این روش صرفه جویی در هزینه های ناشی از ارسال جداگانه بسته های حمل کننده مورچه ها است. از آنجا

¹ Piggyback

که بسته های حامل مورچه ها اندازه های کوچکی دارند و در هنگام ارسال بیشتر قسمت های آنها به جای اطلاعات با سرباره های ارسال پر می شود، همراه شدن با پیام های به روز رسانی این هزینه اضافی را حذف می کند. مزیت دیگر همراه کردن مورچه ها با پیام های به روز رسانی آن است که مورچه ها بیشتر از مسیرهایی عبور خواهد کرد که حجم بیشتری از ترافیک در آنها جریان دارد. بنابراین بهینه سازی تأخیر در مسیرهای پر ازدحام کارایی بیشتری را خواهد داشت که در کمینه سازی تابع هزینه کلی موثرتر است. نقاط شبکه overlay مسئول تولید، انتقال و پردازش اطلاعات مورچه ها هستند. در واقع نقاط با کمک مورچه ها اطلاعات لازم برای تخمین هزینه پیوند های شبکه را جمع آوری می کنند.

زمانی که هر نقطه شبکه N_d ، بخواهد یک پیام به روز رسانی به نقطه دیگر N_s بفرستد، مورچه ای را نیز تولید می کند که فرمت آن به صورت (N_d_Id, N_s_Id, C) می باشد که C هزینه مسیر بین نقطه دریافت کننده بسته و N_d است. مقدار C در ابتدای تولید مورچه صفر بوده و در هر نقطه میانی به اندازه تأخیر میان آن نقطه با نقطه قبلی اضافه می شود. همزمان با عبور مورچه (N_d_Id, N_s_Id, C) از نقطه مبدا تا مقصد، جدول مسیر یابی تمام نقاط میانی به روز رسانی می شود. قوانین به روز رسانی جدول های مسیریابی مشابه با نسخ اصلی الگوریتم بهینه سازی با گروه مورچه ها می باشد [81].

در الگوریتم پیشنهادی به جای استفاده از یک جدول مسیر یابی با عضو های ثابت، هر نقطه یک جدول مسیر یابی با عضو های متعدد را نگه داری می کند که انتخاب میان عضو ها با قوانین احتمال صوت می گیرد. به این جدول مسیر یابی در اصطلاح جدول مسیریابی احتمالی گفته می شود. بنابراین در کنار هر عضو جدول مسیر یابی عددی به عنوان احتمال انتخاب آن در میان سایر کاندیداهای موجود، نگه داری می شود که این اعداد به صورت مداوم توسط مورچه ها به روز رسانی می شود. از آنجا که مورچه ها دائماً در شبکه در حال حرکت هستند، نرخ به روز رسانی جدول های مسیریابی با توجه به

شرایط دینامیک شبکه های محیط های مجازی می تواند به اندازه کافی زیاد باشد. شکل ۴-۸ جدول مسیر یابی پیشنهادی را نشان می دهد که فرم اصلاح شده ای از جدول مسیر یابی Pastry می باشد. همانطور که در شکل دیده می شود در هر خانه جدول مسیر یابی تعداد پنج انتخاب با شناسه مناسب نگهداری می شود. مقادیر برخی از خانه ها در شکل برای نمونه پر شده است و سایر خانه های هاشور زده شده خانه های مشابهی با مقادیر مشابه را نشان می دهند و خانه های خالی مکان هایی هستند که همسایه مناسبی برای آنها پیدا نشده است.

سطرهای پایینی جدول مسیر یابی متناظر با تعداد شناسه های محدودی هستند، در حالیکه سطرهای بالاتر جدول به روز رسانی محدوده وسیعی از شناسه ها را در بر می گیرند. بنابراین تعداد محدود کاندیداهای نگه داری شده در هر خانه جدول مسیر یابی، در سطرهای پایین درصد قابل توجهی از انتخاب های موجود را شامل می شود. بنابراین الگوریتم مسیر یابی برای یافتن همسایه های نزدیک در سطرهای پایینی جدول مسیر یابی با کارایی بیشتری عمل می کند. نکته قابل توجه این است که سطرهای پایینی جدول مسیریابی متناظر با همسایه های نزدیک در محیط مجازی است که تقاضای ارسال ترافیک به آنها بیشتر بوده و اولویت کمینه کردن تأخیر برای آنها بیشتر است.

Leaf set																			
0-2212102	0-3120203	0-2013023	0-1121302	0-1121302	1														
0										1-2-301203	1-2-231203	1-2-221331	1-2-021302	1-2-301230					
10-0-03201	10-0-31230	10-0-45620	10-0-12301	10-0-12301						2									
										102-2-2302	102-2-1230	102-2-1020	102-2-3232	102-2-0000	3				
					1023-1-000	1023-1-231	1023-1-331		1023-1-123						3				
										10233-2-32	10233-2-13								
0										102331-2-0									
										2									

شکل ۸-۴ جدول مسیریابی ساختار DHT

هر نقطه ساختار به جای نگه داری آدرس تنها یک نقطه در هر خانه جدول مسیریابی، ساختاری را به صورت $((y_1, p_1), (y_2, p_2), (y_3, p_3), \dots, (y_k, p_k))$ در هر خانه جدول مسیریابی تشکیل می دهد که در آن $p_1, p_2, \dots, p_k$ احتمالات متناظر با انتخاب $y_1, y_2, \dots, y_k$ می باشد. در واقع فرومون در مسیریابی گروه مورچه ها در مسأله ما با احتمالات $p_1, p_2, \dots, p_k$ نمایش داده می شود. نکته قابل توجه آن است که هر نقطه باید مسیر بهینه متناظر با تمام خانه های جدول مسیریابی را پیدا کند. بنابراین در اینجا فرومون های مختلفی برای خانه های مختلف جدول مسیریابی وجود دارد. زمانی که مورچه (N_d_Id, N_s_Id, C) از طریق پیوند L به یک نقطه می رسد، هزینه C به اندازه تأخیر پیوند L در جهت معکوس اضافه می شود. پس از آن نقطه دریافت کننده مورچه کاندیداهای $(y_1, y_2, \dots, y_k)$ ارسال نقطه N_d_Id را در جدول مسیریابی خود پیدا می کند. اگر نقطه N_d به

مجموعه این کاندیداها تعلق داشته باشد ($\exists i, y_i = N_d \text{Id}$) احتمال متناظر با $(y_1, y_2, \dots, y_k)$ بر طبق روابط زیر به روز رسانی می شود :

$$P_i = \frac{P_i^{old} + \Delta P}{1 + \Delta P} \quad (4-4)$$

$$P_j = \frac{P_j^{old}}{1 + \Delta P} \quad (5-4)$$

$$\Delta P = K / f(C) \quad (6-4)$$

که در آن P_i^{old} مقدار فعلی P_i برای y_i است و ΔP تابعی غیر صعودی از C است.

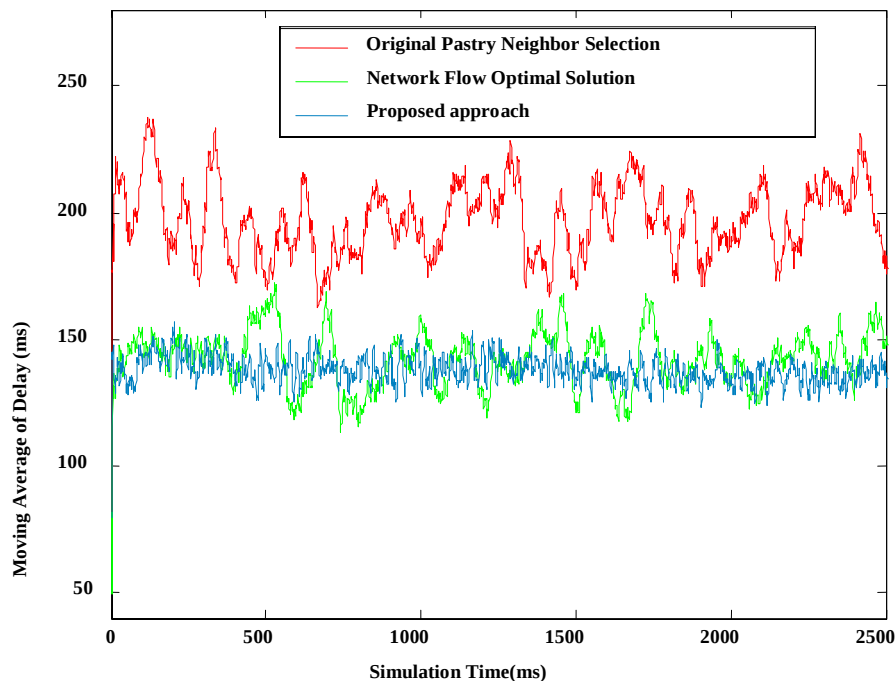
در شرایطی که نقطه N_d به مجموعه $(y_1, y_2, \dots, y_k)$ تعلق نداشته باشد، N_d جایگزین y_i می شود که کوچکترین مقدار p_i را داشته باشد. احتمال های مربوط سایر $k - 1$ عضو بر مبنای روابط قبلی محاسبه می شود.

همانطور که قبلاً هم اشاره کردیم، مورچه ها بر روی پیام های به روز رسانی سوار می شوند و بنابراین با نرخ بالایی در شبکه در حرکت بوده و جدول های مسیر یابی نقاط را به روز رسانی می کنند.

۴.۴. نتایج شبیه سازی

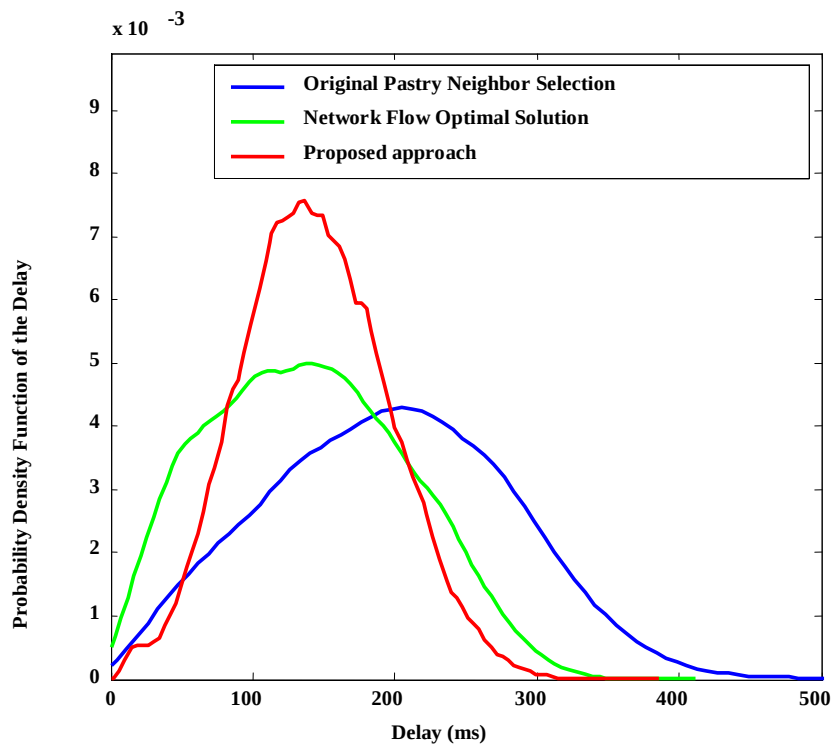
در این قسمت ما نتایج شبیه سازی الگوریتم مسیریابی پیشنهادی را ارایه خواهیم داد. نتایج حاصل از شبیه سازی سیستم پیشنهادی در شبیه ساز FreePastry می باشد که یک پیاده سازی با کد قابل دسترسی از Pastry است. یکی از ورودی های این شبیه ساز ماتریس تأخیرهای میان نقاط شبکه اینترنت است. بنابراین شبیه سازی توپولوژی اینترنت در حوزه شبیه ساز FreePastry قرار نمی گیرد و به عنوان ورودی به آن داده می شود. در شبیه سازی های ارایه شده در این فصل نتایج حاصل از روش

پیشنهادی با ساختار اصلی Pastry و همچنین پاسخ بهینه حاصل از حل مسأله بهینه سازی قسمت قبل به دست آمده است، مقایسه می شود. مقایسه با Pastry از آن جهت است که ساختار پیشنهادی بر مبنای شبکه Pastry طراحی شده است. ولی نکته قابل توجه این است که پس از Pastry گروه وسیعی از تحقیقات ساختارهای مختلفی را برای محیط های مجازی پیشنهاد داده اند که پیاده سازی و مقایسه آنها با روش پیشنهادی به دلیل در دسترس نبودن جزئیات آنها و شرایط شبیه سازی ارایه شده و تعداد آنها امکان پذیر نمی باشد. از این رو ما پاسخ بهینه مسأله را به عنوان کران بالای بهینه پاسخ های بهینه در نظر گرفته و کارایی الگوریتم پیشنهادی را با آن مقایسه خواهیم نمود. شبیه سازی ها در یک محیط مجازی با ۱۰۰۰۰ کاربر و شرایط ذکر شده در فصل سوم انجام شده است. برای تخصیص شناسه به کاربران از منحنی هیلبرت مرتبه ۱۰ استفاده شده است. شکل ۴-۹ میانگین در حرکت تأخیر را برای ترافیک در هر سه حالت نشان می دهد.



شکل ۴-۹ میانگین در حرکت تأخیر (ms) برای شبکه Pastry، روش پیشنهادی و ساختار بهینه

نکته قابل توجه در شکل ۴-۹ آن است که به علت ماهیت تصادفی ترافیک نتایج لحظه ای تأخیر ممکن است گاهی در روش پیشنهادی ما نسبت به روش بهینه کمتر باشد در حالیکه در میانگین روش بهینه بهترین کارایی را دارا می باشد. شکل ۴-۱۰ تابع توزیع احتمال تأخیر را در هر سه حالت نشان می دهد. همانطور که در شکل دیده می شود، تأخیر شبکه Pastry مقادیر تأخیرهای بالا تا ۵۰۰ msec را شامل می شود در حالی که تابع توزیع تأخیر در حالت بهینه و همچنین برای روش پیشنهادی ما، توزیع تأخیر به ۳۰۰ msec محدود می شود. علاوه بر آن در بیشتر موارد تأخیر در حوزه ۱۰۰ msec تا ۲۰۰ msec قرار دارد که نیازهای کاربرد بلادرنگ محیط های مجازی را برآورده می کند [82].



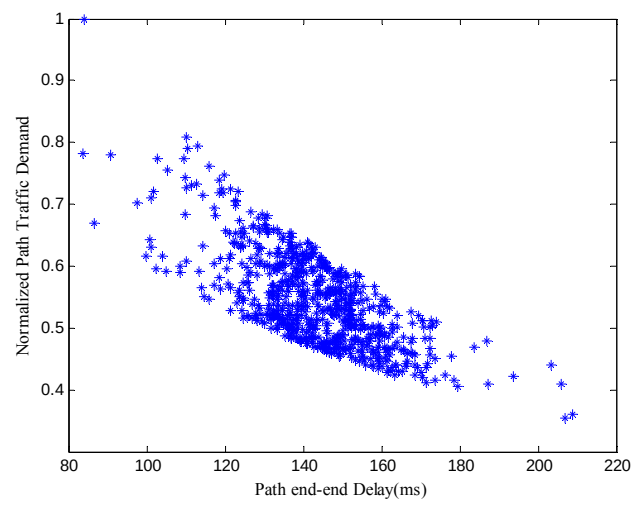
شکل ۴-۱۰ تابع توزیع تأخیر (ms) برای شبکه Pastry، روش پیشنهادی و ساختار بهینه

نکته قابل توجه دیگر در شکل ۴-۱۰ این است که توزیع تأخیر در روش پیشنهادی ما نسبت به دو روش دیگر واریانس کمتری دارد. واریانس کمتر سبب کاهش لغزش در ارسال ترافیک خواهد شد که

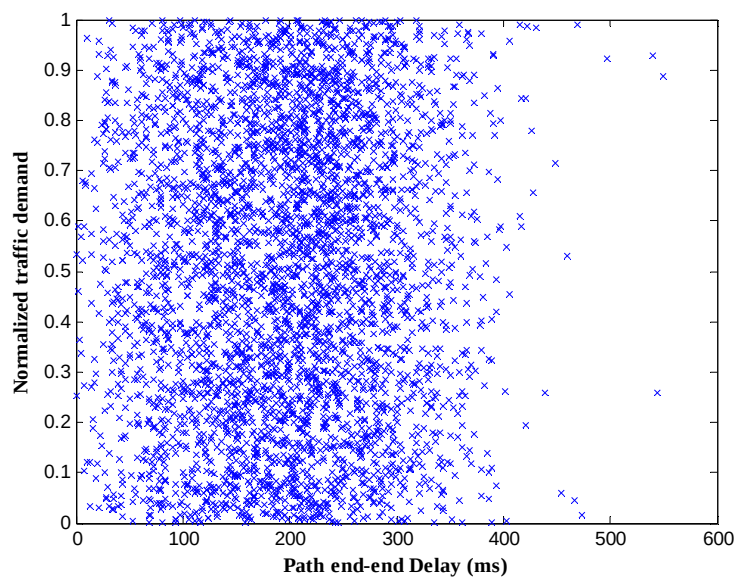
در کاربر محیط های مجازی بسیار مطلوب می باشد. تحقیقات نشان داده است که در محیط های مجازی شبکه ای تأثیر نامطلوب لغزش در همکاری از تأثیر تأخیر بیشتر است [82].

شکل ۴-۹ میانگین در حرکت تأخیر را برای یک نقطه در شبکه نشان می دهد، در حالیکه شکل ۴-۱۰ تابع توزیع تأخیر را در کل شبکه نشان می دهد. بنابراین واریانس تأخیر در شکل ۴-۱۰ نسبت به شکل ۴-۹ بیشتر است. تفاوت میان این واریانس ها در شبکه Pastry که نقاط در روی Overlay به صورت تصادفی پخش شده اند، به مراتب بیشتر می باشد. بنابراین برخی از بسته ها باید به نقاط نزدیک و برخی دیگر باید به نقاط دور فرستاده شوند. در حالیکه در ساختار پیشنهادی بیشتر بسته ها وضعیت مشابهی را دارا هستند.

یکی دیگر از اهداف طراحی ما بهینه سازی تأخیر بر مبنای تقاضای ترافیک می باشد. این بدان معنا است که در بهینه سازی تأخیر اولویت به مسیرهای پر ازدحام داده شود. برای نشان دادن کارایی الگوریتم پیشنهادی در این زمینه، شبیه سازی دیگری انجام شده است که در آن مقادیر متوسط تأخیر در مسیر های مختلف شبکه و میزان تقاضای ترافیک در آن مسیر ها ثبت شده است. میزان تقاضای ترافیک برای یک مسیر متوسط ترافیک عبوری از آن مسیر در یک بازه از زمان است. برای نرمالیزه شدن مقادیر در بازه [۱۰]، تقاضای مسیر ها را به مقدار ماکزیمم تقاضا در شبکه تقسیم شده است تا مقادیر تقاضای نرمال به دست بیاید. شکل ۴-۱۱ تأخیر مسیرهای شبکه را در مقابل تقاضای ترافیک در آنها نشان می دهد و شکل ۴-۱۲ پارامتر مشابه را برای شبکه Pastry نشان می دهد. نتایج شکل ۴-۱۱ به روشنی این نکته را نشان می دهد که متوسط تأخیر در مسیر هایی که تقاضای ترافیک آنها بیشتر است، کمتر می باشد و این در حالی است که رابطه متوسط ترافیک با تقاضای مسیرها در شکل ۴-۱۲ کاملاً تصادفی است. بنابراین روش پیشنهادی در کمینه سازی تأخیر به مسیر های پر ازدحام اولویت بیشتری داده است.



شکل ۴-۱۱ تأخیر مسیرهای شبکه در مقابل تقاضای ترافیک در آنها در ساختار پیشنهادی



شکل ۴-۱۲ تأخیر مسیرهای شبکه در مقابل تقاضای ترافیک در آنها در شبکه Pastry

فصل پنجم

ساختار مبتنی بر منحنی هیلبرت

اندیس دهی چند بعدی با منحنی هیلبرت

مسیریابی به کمک تبدیل هیلبرت

الگوریتم مسیریابی

نتایج شبیه سازی

همانطور که در فصول قبل اشاره کردیم، با وجود آنکه ساختارهای مختلفی برای محیط های مجازی پیشنهاد شده است، توزیع پیام های به روز رسانی با تأخیر کمینه هنوز هم یکی از چالش های بزرگ در محیط های مجازی گسترده است. در این فصل یک ساختار گسترده دیگر را برای محیط های مجازی ارایه خواهیم داد که با استفاده از الگوی تبادل پیام ها سعی در مسیر یابی آنها از کوتاه ترین راه ها دارد.

پیام های به روز رسانی هر نقطه در یک محیط مجازی معمولاً به ناحیه ای اطراف آن ناحیه که از آن نقطه تأثیر می پذیرند، ارسال می شود. در یک محیط مجازی گسترده هیچ پایگاه داده مرکزی برای نگه داری آدرس IP نقاط موجود در هر ناحیه وجود ندارد. در عدم حضور چنین سرویس مرکزی مکان یابی و برای اجتناب از ارسال درخواست های مکرر در شبکه برای یافتن آدرس IP نقاط موجود در هر ناحیه، در ساختار پیشنهادی مسیر یابی به جای آدرس IP بر مبنای شناسه ای انجام می شود که از روی مکان نقاط قابل استخراج است تا نیاز به سرویس مکان یابی برطرف شود. علاوه بر آن، با انتخاب شناسه

های مناسب برای نقاط بر مبنای موقعیت مجازی آنها شبکه ساخته شده بر مبنای فاصله ها در محیط مجازی شکل خواهد گرفت و تعداد گام های میان هر دو نقطه با فاصله آنها در محیط مجازی نسبت خواهد داشت. بنابراین چنین ساختاری برای ارسال پیام های به روز رسانی که بیشتر به همسایگی های نزدیک ارسال می شوند، مناسب است.

نقاط یک محیط مجازی را می توان به صورت رکوردهای یک پایگاه داده گسترده در نظر گرفت که به طور مداوم به روز رسانی می شوند. رکورد متناظر با هر شیء در محیط مجازی شامل تمامی اطلاعات مرتبط با آن شیء و وضعیت فعلی آن می باشد. هر تغییر در ناحیه مورد علاقه یک شیء منجر به ارسال یک درخواست برای به روز رسانی رکورد آن می شود.

در یک پایگاه داده، هر رکورد با یک یا چند ویژگی آن آدرس دهی می شود که در کاربرد مورد نظر ما این ویژگی ها مختصات x, y, z و نقطه مربوطه می باشند. این انتخاب به آن دلیل است که درخواست های به روز رسانی یک نقطه باید به ناحیه ای از فضا ارسال شود که با مختصات نقاط آن تعریف می شود. بنابراین بهترین مبنا برای آدرس دهی رکوردهای این پایگاه داده، مختصات آنها می باشد. از آنجا که اشیاء محیط مجازی با مختصات سه بعدیشان در فضای مجازی آدرس دهی می شوند، به جای یک ویژگی دارای چند ویژگی مستقل شاخص هستند.

بنابراین توزیع پیام های به روز رسانی به مکان های مختلف محیط مجازی می تواند به صورت اجرای درخواست های چند بعدی^۱ بر روی اندیس های پایگاه داده تعریف شود. اندیس دهی چند بعدی مسأله جستجو در یک پایگاه داده با چند کلید مستقل که در کنار یکدیگر یک رکورد یکتا را مشخص می کنند، را مورد بررسی قرار می دهد. در مسأله محیط مجازی این سه کلید مستقل مختصات

^۱ Multi-dimensional index querying

x, y و z هر نقطه هستند که در کنار یکدیگر یک نقطه یکتا را در فضا مشخص می کنند، در حالیکه اشیاء می توانند مختصات های یکسان در یک یا دو بعد داشته باشند، مختصات هیچ دو نقطه ای در هر سه بعد یکسان نخواهد شد.

همچنین از آنجا که پیام های به روز رسانی به جای تک نقاط به نواحی پیوسته در فضا ارسال می شوند، بیشتر در خواست های این پایگاه داده از نوع درخواست های محدوده ای¹ می باشد که در آن درخواست ارسال شده برای جستجوی رکوردهای موجود در یک محدوده در یک یا چند بعد می باشد. اندیس های چند بعدی در یک پایگاه داده به معنای مشخص کردن رکوردهایی با بیش از یک ویژگی است. بنابراین به جای جستجوی رکوردها با یک کلید، چند کلید که اهمیت یکسان دارند برای جستجو مورد استفاده قرار می گیرند.

به منظور جستجوی اطلاعات در پایگاه های داده چند بعدی باید اطلاعات آنها با روش مناسبی اندیس دهی شود. به عبارت دیگر این اطلاعات باید با نگاشت مناسب از چند بعد به یک بعد نگاشته شوند. منحنی هیلبرت روشی برای نگاشت اطلاعات چند بعدی به یک بعد است. در این رساله برای اندیس دهی چند بعدی از منحنی هیلبرت استفاده می شود. در فصل گذشته در مورد منحنی هیلبرت صحبت کردیم و اشاره نمودیم که منحنی پر کننده فضای هیلبرت از نزدیکی هر نقطه در فضا می گذرد و یک نگاشت را میان مختصات نقاط و شناسه آنها روی منحنی هیلبرت تعریف می کند. انگیزه اصلی استفاده از منحنی هیلبرت خواص مطلب آن در حفظ نسبت فاصله نقاط هنگام نگاشت آنها به فضای تک بعدی شناسه ها می باشد. [84] [83] روش کارآمدی را برای نگاشت اطلاعات چند بعدی به کمک منحنی هیلبرت ارائه داده است که الگوریتم مسیریابی پیشنهادی از ایده های ارائه شده در آن به منظور

¹ Range query

اجرای درخواست های چند بعدی بر روی یک پایگاه داده الهام گرفته است. ایده اصلی روش پیشنهادی در این فصل استفاده از فرآیند مرحله ای نگاشت بر روی منحنی هیلبرت در راستای مسیریابی پیام های به روز رسانی به نقاط مختلف محیط مجازی می باشد. به طوری که پیام های به روز رسانی که آدرس مقصد آنها به شکل یک مختصات سه بعدی تعریف شده است، به سمت نقطه ای با نزدیک ترین شناسه به شناسه هیلبرت مقصد تعیین شده برسد. در فرآیند مسیر یابی، هدف حرکت میان نقاط منحنی هیلبرت در کوتاه ترین مسیرها است. به این منظور ما از ساختار درختی منحنی هیلبرت استفاده نموده و مسیر یابی را بر روی این درخت تعریف می کنیم. یک منحنی هیلبرت مرتبه K با یک درخت با K مرتبه نشان داده می شود. به منظور مسیر یابی بهینه بر روی درخت نقاط تنها تعداد محدودی از همسایگان خود را بر روی درخت به منظور هدایت بسته ها در شاخه های مناسب، بشناسد. از آنجا که بیشتر پیام های به روز رسانی به نقاط نزدیک به هم که شاخه های مشترک دارند، ارسال می شود انتخاب همسایه ها باید به صورتی است که تعداد گامها در این مسیرها کم باشد.

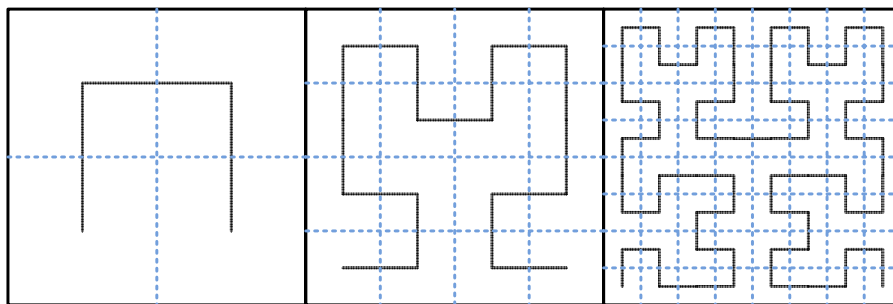
نکته قابل توجه در مورد الگوریتم نگاشت مورد استفاده آن است که این روش یک الگوریتم مرحله ای است که در هر مرحله با توجه به وضعیت تعدادی از بیت های مختصات در مورد خروجی آن مرحله تصمیم گیری می شود. بنابراین، الگوریتم مسیر یابی ارایه شده نیازمند به انجام نگاشت کامل نقاط بر روی منحنی هیلبرت که بار محاسباتی زیادی را دارد، نمی باشد و در هر مرحله از مسیریابی تنها قسمتی از این نگاشت انجام می شود. بنابراین روش پیشنهادی نقاط شبکه را از محاسبه و ذخیره سازی درخت متناظر با جدول هیلبرت که هزینه زیادی را به همراه دارد، بی نیاز می کند.

در قسمت های بعدی این فصل فرآیند مرحله ای که در آن مختصات سه بعدی نقاط به شناسه آنها روی منحنی هیلبرت نگاشته می شود، شرح داده خواهد شد.

۱.۵. اندیس دهی چند بعدی با منحنی هیلبرت

همانطور که قبلاً هم اشاره کردیم، تکنیک مورد استفاده ما در این بخش استفاده از فرآیند نگاشت سه بعدی به کمک منحنی هیلبرت است. برای توصیف بهتر الگوریتم، ابتدا نمای تصویری آن را در دو بعد شرح می‌دهیم و پس از آن جزئیات الگوریتم را در سه بعد توضیح خواهیم داد. منحنی هیلبرت فضا را به صورت بازگشتی به ربع‌هایی تقسیم می‌کند که در شکل ۱-۵ در سه مرتبه متوالی نشان داده شده است.

با استفاده از منحنی هیلبرت فضا به شکل بازگشتی به ربع‌هایی تقسیم می‌شود به طوری که منحنی هیلبرت مرتبه N از تقسیم چهار تایی ربع‌های منحنی هیلبرت مرتبه N به دست می‌آید. هر نقطه روی منحنی هیلبرت یک شماره ترتیب^۱ منحصر به فرد دارد که به این شماره‌های ترتیب در اصطلاح کلید^۲ نقطه مورد نظر نیز گفته می‌شود. هدف نگاشت مرحله‌ای آن است که در تعدادی مرحله و در هر مرحله با استفاده از تعدادی بیت‌های مختصات نقطه مورد نظر، قسمتی از کلید آن نقطه مشخص شود. بنابراین منحنی پرکننده فضای هیلبرت به شیوه‌ای بازگشتی از واحدهای اولیه ساخته



شکل ۱-۵ ربع فضاهای تشکیل شده توسط منحنی هیلبرت

^۱ Sequence Number

^۲ Derived Key

می شود و رابطه میان نقاط روی منحنی هیلبرت و شماره ردیف تخصیص داده شده به آنها به صورت یک ساختار درختی قابل نمایش است که ارتفاع این درخت برابر با مرتبه منحنی هیلبرت است.

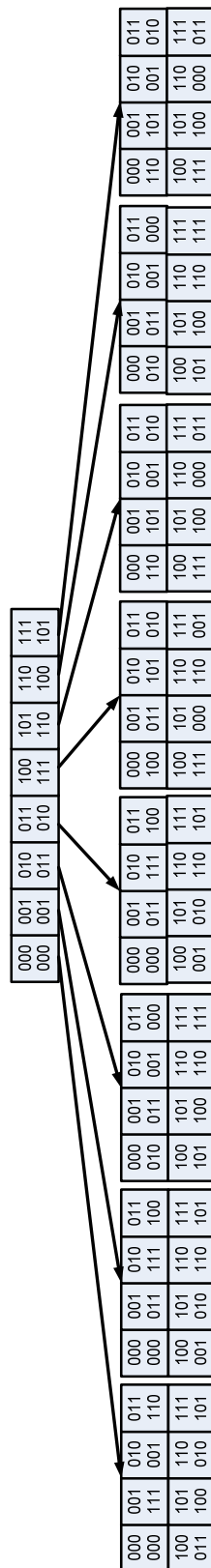
اگر نقاط منحنی هیلبرت به شکل یک درخت آرایش یافته باشند، برای یافتن مکان یک نقطه روی آن درخت، الگوریتم از ریشه درخت آغاز می شود و در هر مرحله یک شاخه به نقطه مورد نظر روی منحنی نزدیک تر می شود. ساختار درختی منحنی هیلبرت در [83] ارایه شده است. هر نقطه درخت متناظر با منحنی هیلبرت سه بعدی شامل ۸ زوج از نقاط است که هر یک از این زوج ها یک کلید و مختصات متناظر با آن را مشخص می کند. ارتباط میان این کلیدها و مختصات ها را نوع نقطه مشخص می کند. یک درخت هیلبرت سه بعدی ۱۲ نوع نقطه دارد که برای هر نوع از این نقاط این ۸ زوج به صورت مشخص تعریف می شوند. هر نقطه درخت به جز برگ های آن، مادر ۸ نقطه دیگر درخت خواهد بود که هر یک از این ۸ نقطه در واقع از یکی از ۸ زوج داخل نقطه مادر منشعب می شوند. نوع هر یک از این ۸ نقطه بسته به حالت نقطه مادر و وضعیت زوج بالای آن دارد که این ارتباط با یک نمودار حالت مشخص می شود. حالت های این نمودار در جدول ۵-۱ نشان داده شده است. همانطور که در این جدول دیده می شود، این نمودار ۱۲ حالت دارد که هر حالت متناظر با یکی از انواع نقاط درخت هیلبرت است. ریشه درخت یک نقطه نوع صفر است و پس از آن درخت بر طبق این نمودار حالت گسترش می یابد.

جدول ۱-۵ ارتباط میان مختصات نقاط و کلید آنها در هر حالت

state no.	<i>derived-key</i>							
	000	001	010	011	100	101	110	111
0	000	001	011	010	110	111	101	100
	1	2	2	3	3	5	5	4
1	000	010	110	100	101	111	011	001
	2	0	0	8	8	7	7	6
2	000	100	101	001	011	111	110	010
	0	1	1	9	9	11	11	10
3	011	010	000	001	101	100	110	111
	11	6	6	0	0	9	9	8
4	101	111	011	001	000	010	110	100
	9	7	7	11	11	0	0	5
5	110	010	011	111	101	001	000	100
	10	8	8	6	6	4	4	0
6	011	111	110	010	000	100	101	001
	3	11	11	5	5	1	1	7
7	101	100	110	111	011	010	000	001
	4	9	9	10	10	6	6	1
8	110	100	000	010	011	001	101	111
	5	10	10	1	1	3	3	9
9	101	001	000	100	110	010	011	111
	7	4	4	2	2	8	8	3
10	110	111	101	100	000	001	011	010
	8	5	5	7	7	2	2	11
11	011	001	101	111	110	100	000	010
	6	3	3	4	4	10	10	2

شکل ۲-۵ درخت هیلبرت سه بعدی ساخته شده بر طبق نمودارهای هیلبرت ارائه شده در قبل

را نشان می دهد.



شکل ۲-۵ نمایش درختی منحنی سه بعدی هیلبرت

نمایش درختی منحنی هیلبرت علاوه بر نمایش سلسله مراتبی فرآیند نگاشت، می تواند گویای الگوریتم مناسب برای یافتن کلید متناظر با یک مختصات سه بعدی باشد. این الگوریتم می تواند به صورت مرحله ای اجرا شود که شبه کد آن در شکل ۳-۵ آمده است.

```

current_level <= 1
current_state <= 0
D <= ""
Repeat
    P <= Concat( Xp(current_level), Yp(current_level), Zp(current_level))
    d <= derived_key(P, current_state)
    D <= concat(D, d)
    If current_level < curve_order then
        current_state <= next_state(current_state, P, d)
    end if
    current_level <= current_level + 1
Until current_level > curve_order
    
```

شکل ۳-۵ شبه کد نگاشت مرحله ای نقاط روی منحنی هیلبرت

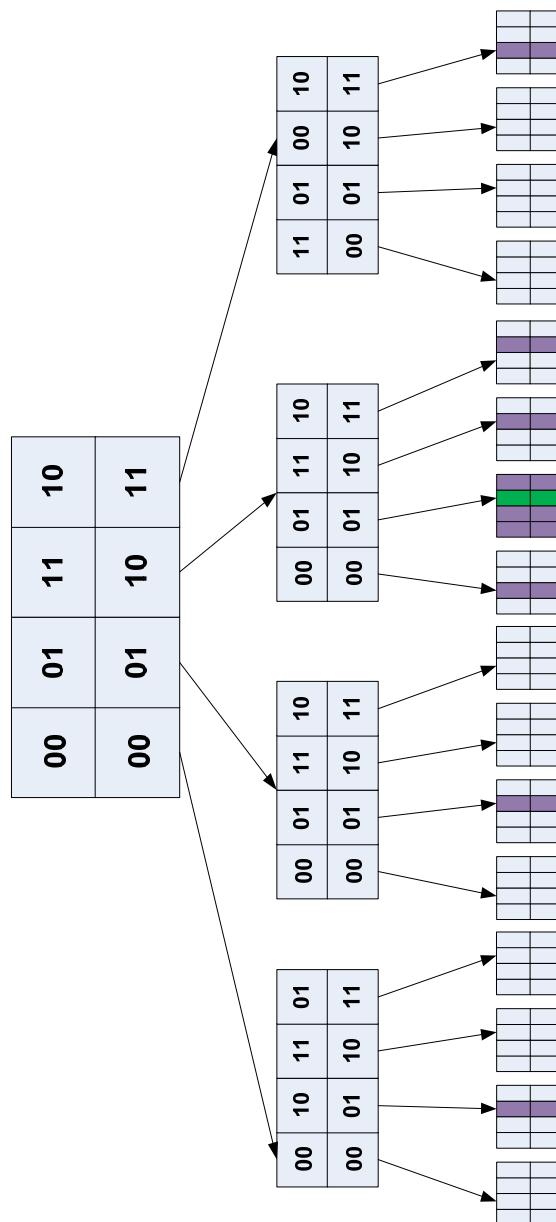
۲.۵. مسیریابی به کمک تبدیل هیلبرت

در این قسمت با الهام از روش تبدیل مرحله ای شرح داده شده در قبل، به توصیف الگوریتم مسیریابی پیشنهادی خواهیم پرداخت. در الگوریتم مسیریابی پیشنهادی، هر بسته با عبور بر روی شاخه های درخت در هر گام یک شاخه به مقصد نزدیک می شود. به این صورت بسته میان نقاط مختلف درخت هیلبرت منتقل می شود و در سلسله مراتب پایین می رود تا به نقطه مورد نظر خود برسد. برای انتقال مناسب بسته های دریافتی، هر نقطه باید اطلاعاتی از نقاط درخت را در یک جدول مسیریابی نگه داری کند. جدول مسیریابی برای یک منحنی هیلبرت سه بعدی از مرتبه K دارای K سطر و ۷ ستون

است. علاوه بر آن مجموعه کوچکی از نزدیکترین نقاط به هر نقطه نیز در جدول مسیریابی آن به عنوان مجموعه برگ ها نگهداری می شود. هر خانه جدول مسیریابی جایگاه آدرس IP نقطه ای است که در زیر شاخه معینی از درخت قرار گرفته است. ۷ عضو موجود در سطر N ام جدول مسیریابی هر نقطه به همسایه ای که در $N-1$ رده از شاخه ها با نقطه مورد نظر مشترک است و در شاخه مرتبه N ام از آن جدا می شود، اشاره می کند. علاوه بر آن، مجموعه برگ های جدول مسیریابی هر نقطه نزدیک ترین همسایگان شناخته شده توسط آن نقطه هستند. نقاط این مجموعه در زمان هایی که بسته امکان انتقال به یک نقطه نزدیک تر به مقصدش را ندارد مورد استفاده قرار می گیرد. در چنین شرایطی بسته به نزدیکترین نقطه به مقصد در مجموعه برگ ها ارسال می شود.

یک درخت هیلبرت مرتبه K به تعداد 2^{3K} برگ خواهد داشت و هر یک از ۸ نقطه مرتبه اول این درخت خود 2^{3K-3} زیر شاخه دارند. در سطر اول جدول مسیر یابی، هر نقطه یک نماینده از میان هر گروه از این 2^{3K-3} زیرشاخه برای خود پیدا می کند. زیر شاخه های مرتبه دوم درخت هیلبرت هر یک 2^{3K-6} نقطه در زیر خود دارند. هر نقطه در سطر دوم جدول مسیر یابی خود تنها نماینده هایی از شاخه های مرتبه دومی را که شاخه مرتبه اول آنها با نقطه مربوطه یکسان است، نگهداری می کند. به صورت مشابه در میان شاخه های مرتبه سوم که هر یک 2^{3K-9} نقطه در زیر خود دارند، هر نقطه تنها از میان شاخه های مرتبه سومی که در شاخه مرتبه ۲ با آن یکسان هستند، یک نماینده انتخاب می کند. در نهایت خانه های سطر آخر جدول مسیر یابی هر یک به عضو یکتایی در روی درخت اشاره می کنند که در تمام شاخه ها با نقطه مورد نظر یکسان است. نکته قابل توجه آن است که با توجه به محدود بودن انتخاب ها برای همسایه های سطرهای پایینی جدول مسیریابی، برخی از این در سطرهای آخر ممکن است خالی بمانند. که برای این حالت ها در جدول مسیر یابی تدابیر لازم اندیشیده می شود. به دلیل زیاد

بودن تعداد نقاط درخت هیلبرت سه بعدی ، همسایگی های توضیح داده شده برای درخت هیلبرت دو بعدی مرتبه ۲ در شکل ۴-۵ به تصویر درآمده است. در این شکل نقطه مورد نظر با رنگ سبز و همسایه های آن با رنگ بنفش مشخص شده اند. شکل ۵-۵ نیز جدول مسیر یابی نقطه ای با کلید ۶۰۰ را در یک منحنی هیلبرت سه بعدی نشان می دهد.



شکل ۴-۵ نمونه ای از تعریف همسایگی ها برای یک نقطه درخت دو بعدی هیلبرت

Routing Table for Object with key 600

Routing Set:

Branch	0	1	2	3	4	5	6	7
Order								
1	21	--	1300	1800	2300	2900	3400	4000
2	550	--	670	750	810	860	940	1010
3	580		595	--	611		628	637
4	--	601					606	

Leaf Set:

601	606	611	618
565	580	585	595

شکل ۵-۵ جدول مسیریابی ساختار هیلبرت

۳.۵. الگوریتم مسیریابی

در ساختار پیشنهادی آدرس بسته ها به شکل مختصات مقصد در محیط مجازی مشخص می شود و هر نقطه با در دست داشتن جدول مسیریابی ذکر شده، الگوریتم مسیریابی پیشنهادی را برای پیدا کردن نقطه مناسب را برای ارسال یک بسته اجرا می کند. توصیف این الگوریتم در شکل ۵-۶ آمده است. در توصیف الگوریتم فرض بر آن است که نقطه X بسته ای را به مقصد نقطه P دریافت می کند و در طی مراحل طی در مورد انتقال بسته تصمیم گیری می کند. نمادهای زیر در توصیف شبه کد به کار رفته اند.

K : مرتبه منحنی هیلبرت

$R(i, j)$: عضو جدول مسیریابی در ستون $j \{1 \leq j \leq 7\}$ و سطر $i \{1 \leq i \leq K\}$

$derived_key(m, current_state)$: کلید متناظر با رشته سه بیتی P در حالت مشخص شده با

$current_state$

$next_state(current_state, d)$: حالت بعدی دیاگرام متناظر با منحنی هیلبرت نشان داده

شده در جدول ۵-۱ در شرایطی که حالت فعلی با $current_state$ مشخص می شود و کلید متناظر در این حالت d است.

الگوریتم پیشنهادی قسمت های اصلی زیر را در مراحل متعدد تکرار می کند تا عضو مناسب را در جدول مسیر یابی پیدا شود و اگر این عضو در جدول مسیر یابی پیدا نشود به نزدیک ترین عضو در اجتماع مجموعه برگ ها و نقطه X ارسال می شود.

- در هر مرحله K امین بیت از سمت MSB هر یک از مختصات های x, y و z نقطه P به هم چسبانده می شود و یک عدد سه بیتی ساخته می شود. کلید متناظر با این نقطه سه بعدی با توجه به وضعیت این سه بیت و حالت فعلی منحنی هیلبرت محاسبه می شود و حالت فعلی روی منحنی با توجه به این بیت ها به روز رسانی می شود.

- اگر کلید به دست آمده در این مرحله با بیت های متناظر در کلید X تطابق داشته باشد و به عبارت دیگر تا مرحله فعلی نقطه P در تمام شاخه ها با نقطه X مشترک باشد، الگوریتم به مرحله بعدی خواهد رفت تا شاخه های مرتبه پایینتر را بررسی کند. اگر الگوریتم به پایین ترین لایه درخت برسد و تمام کلید های به دست آمده با بیت های متناظر در کلید X مساوی باشند، بسته به X ارسال خواهد شد.

```

current_level = 0
current_state = 0
done_flag = 0
Repeat
    m <= Concat(Xp(K-current_level), Yp(K-current_level), Zp(K-current_level))
    d <= derived_key(m, current_state)
    If ( d = X_key(3(K-current_level) .. 3(K-current_level)-2) ) then
        If ( current_level < K-1 ) then
            current_state <= next_state(current_state, d)
            current_level = current_level + 1
        Else
            Route to ( X ) // the packet is destined for the node X itself
            done_flag = 1
        end if
    Else
        If ((R(current_level+1, d) != null) then
            Route to (R(current_level+1, m))
        Else
            Route to ( L ) st. L ∈ Leaf_set ∪ X, |L-P| : min
        end if
        done_flag = 1
    end if
Until done_flag = 1

```

شکل ۶-۵ شبه کد الگوریتم مسیریابی بر مبنای منحنی هیلبرت

اگر کلید به دست آمده در قسمت قبل با بیت های متناظر در کلید X تطابق نداشته باشد، بسته به خانه ای از جدول مسیر یابی که شماره سطر آن با شماره مرحله فعلی معین می شود و ستون آن طوری انتخاب می شود که کلید به دست آمده در این مرحله با بیت های متناظر در کلید آن تطابق داشته باشد.

- اگر خانه مورد نظر در جدول مسیریابی خالی باشد و به عبارت دیگر در جدول مسیریابی نقطه

مناسبی برای ارسال بسته پیدا نشده است، بسته به عضوی در اجتماع مجموعه برگ ها و X

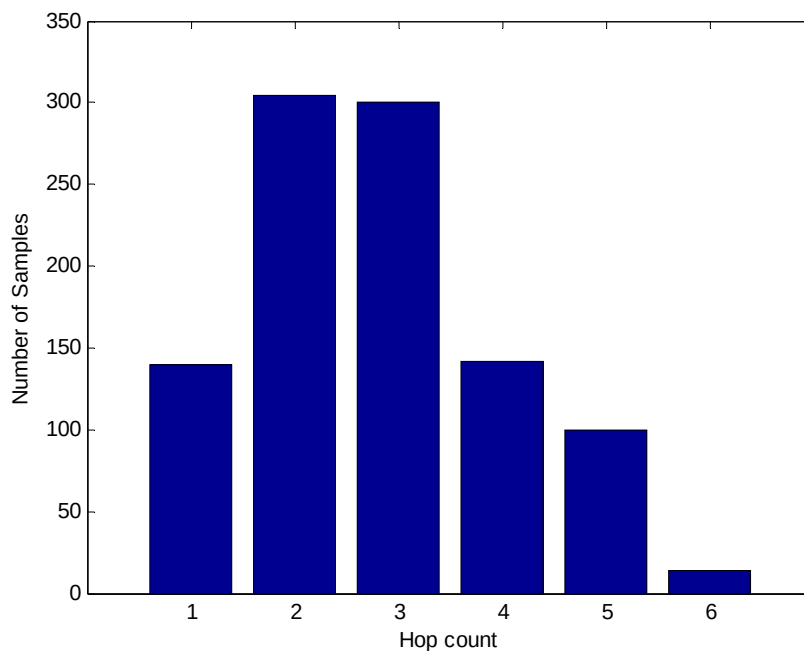
ارسال می شود که کمترین فاصله را از P داشته باشد.

نکته قابل توجه آن است که برای اجرای الگوریتم بالا نقاط تنها نیاز به نگهداری جدول مسیریابی خود و دیاگرام حالت منحنی هیلبرت دارند و نیازی به ذخیر کامل درخت متناظر با منحنی هیلبرت وجود ندارد.

۴.۵. نتایج شبیه سازی

در این قسمت ما نتایج شبیه سازی الگوریتم مسیریابی پیشنهادی بر مبنای منحنی هیلبرت را ارائه خواهیم داد. شبیه سازی ها در یک محیط مجازی گسترده با ۱۰۰۰۰ نقطه و در شرایط تعریف شده در فصل ۳ انجام شده است. برای تخصیص شناسه به کاربران از منحنی هیلبرت مرتبه ۱۰ استفاده شده است. شکل ۵-۷ نتایج شبیه سازی تعداد گام ها را در شرایط ذکر شده نشان می دهد. در این شکل متوسط تعداد گام ها در ارسال بسته ها ۲.۸ است و بیشترین تعداد گام ۶ است. در حالیکه بیشتر پیام ها در کمتر از ۴ گام به مقصد رسیده اند. همچنین در شکل دیده می شود که تعداد بسته هایی که در یک گام به مقصد رسیده اند کمتر از تعداد بسته هایی است که در دو گام به مقصد رسیده اند. این بدین دلیل است که حجم با توان سوم شعاع متناسب است و بنابراین مساحت ناحیه ای که از نقطه ارسال پیام دو گام فاصله دارد، بیشتر از مساحت ناحیه ای با فاصله یک گام است و تعداد نقاط درون آن نیز به نسبت بیشتر است. تعداد بسته هایی که در سه گام به مقصد رسیده اند نیز به دلیل مشابه از تعداد بسته هایی که در دو گام به مقصد رسیده اند بیشتر است. ولی از آنجا که بیشتر همسایه های موجود در ناحیه مورد علاقه یک نقطه در فاصله کمتر از سه گام از آن قرار داشته اند، تعداد پیام های ارسال شده که چهار گام یا بیشتر را در مسیر خود طی کرده اند به میزان قابل توجهی کاهش یافته است.

با توجه به تعداد گام ها در مسیر انتقال بسته ها، تأخیر انتقال یکی از مسایل قابل بحث در الگوریتم پیشنهادی است. یکی از ایرادهای بزرگ وارد شده به سیستم های نظیر به نظیر ایجاد تأخیر در اثر مسیر یابی در چند گام است. نکته قابل توجه این است که بزرگترین انگیزه در استفاده از سیستم های نظیر به نظیر امکان قابلیت توسعه در این سیستم ها است. از آنجا که شبکه های نظیر به نظیر در انجام وظایف مختلف، از امکانات و منابع داخلی شبکه استفاده می کنند، توسعه ابعاد شبکه نیازی به افزایش منابع سرویس دهنده به آن را ندارد. با در نظر گرفتن این ویژگی مطلوب در شبکه های نظیر به نظیر، مقدار متوسط گام ۲.۸ قابل قبول به نظر می رسد و این در حالی است که در ساختارهای مشترک-سرور نیز به دلیل سویچ ترافیک از طریق سرور همیشه دو گام در مسیر بسته ها وجود دارد. علاوه بر آن میزان تأخیر گام های میان مشترکین تا هر سرور ثابت بوده و به علت محدود بودن تعداد سرورها کنترل زیادی روی آن وجود ندارد و این در حالی است که در شبکه های نظیر به نظیر با قابلیت انعطاف بیشتری

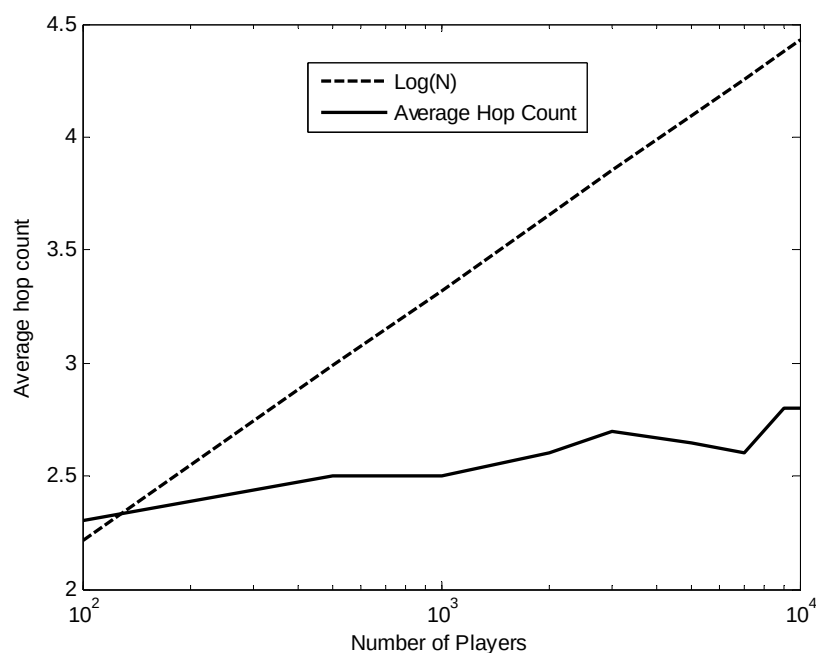


شکل ۵-۷ هیستوگرام گام های ارسال بسته ها

امکان انتخاب گام هایی با تأخیر کمتر را دارند. بنابراین با انتخاب مناسب همسایگی ها در ساختار نظیر به نظیر حتی امکان کاهش تأخیر به مقادیر کمتر از ساختار مرکزی نیز وجود دارد.

۱.۴.۵. قابلیت توسعه

در این قسمت ارتباط میان گام های مسیر یابی را با تعداد کاربران محیط مجازی بررسی خواهیم نمود. به این منظور تعداد کاربران محیط مجازی از ۱۰۰ تا ۱۰۰۰۰ تغییر داده شده است و ابعاد محیط مجازی نیز به نسبت آن تنظیم شده است به طوری که میانگین تعداد کاربران در واحد سطح ثابت بماند. مرتبه منحنی هیلبرت برای حالت ۱۰۰۰۰ کاربر مساوی ۱۰ قرار داده شده است و در سایر موارد به نسبت لگاریتم تعداد کاربران مقدار کمتری برای آن انتخاب شده است. در هر مرحله از شبیه سازی مقدار متوسط تعداد گام های مسیر در شبکه محاسبه و ثبت شده است. شکل ۵-۸ متوسط تعداد گام ها را در هر مرحله در مقابل تعداد نقاط شبکه در آن مرحله نشان می دهد.



شکل ۵-۸ متوسط تعداد گام های مسیر بسته ها بر حسب تعداد نقاط شبکه

همانطور که در نتایج شبیه سازی دیده می شود، ساختار پیشنهادی تعداد گام های ارسال بسته ها را در حد قابل قبولی کوچک نگه می دارد. نکته قابل توجه آن است که الگوریتم های جستجوی همسایه های نزدیک و مسیرهایی نزدیک می توانند مشابه فصل قبل و با تغییرات جزئی در کنار ساختار مبتنی بر منحنی هیلبرت مورد استفاده قرار بگیرند. به دلیل شباهت زیاد این الگوریتم ها به الگوریتم های ذکر شده در فصل قبل، ما از تکرار آنها در اینجا پرهیز می کنیم و تنها نتایج شبیه سازی مربوط به تعداد گام ها را ارائه می دهیم در حالیکه کمینه سازی طول گام ها از لحاظ تأخیر با روش های ذکر شده قابل انجام است.

فصل ششم

ساختار مبتنی بر درخت KD

منحنی پر کننده فضای ترتیب Z

درخت های جستجوی چند بعدی

نمایش منحنی Z در قالب درخت KD

الگوریتم مسیر یابی

نتایج شبیه سازی

در مقدمه فصل قبل چگونگی تعریف مسأله تبادل پیام های به روز رسانی در یک محیط مجازی را در قالب ارسال درخواست هایی به رکوردهای یک پایگاه داده شرح دادیم که این رکورد ها متناظر با اشیاء محیط مجازی هستند و هر تغییر در ناحیه مورد علاقه یک شی منجر به ارسال یک درخواست برای به روز رسانی رکورد آن می شود. همچنین شرح دادیم که ویژگی مورد استفاده در آدرس دهی رکوردها مختصات x, y و z نقطه متناظر با آنها می باشند. از این رو هدف ما در طراحی یک الگوریتم مسیریابی بهینه در واقع حل مسأله جستجوی سریع میان رکوردهای یک پایگاه بزرگ به کمک چندین اندیس است. اندیس دهی به اطلاعات چند بعدی مختصات نقاط در یک پایگاه داده و جستجو بر اساس مجموعه اطلاعات در تمام ابعاد از مراحل مهم در حل مسأله مورد نظر می باشد.

در سال های اخیر تحقیقات گسترده ای در زمینه اندیس دهی و جستجو میان اطلاعات چند بعدی انجام شده است. در این ساختارها هر رکورد به جای یک اندیس با چندین اندیس مشخص می شود و تمام این اندیس ها به میزان مساوی در مکان یابی رکورد تأثیر دارند. مجموعه کلید های

مشخص کننده هر رکورد در پایگاه داده، منحصر به فرد و مشخص کننده آن رکورد است در حالیکه هر یک از کلید ها به تنهایی منحصر به رکورد مورد نظر نمی باشد.

در مورد اشیاء محیط مجازی نیز نکته مورد توجه آن است که یک مختصات خاص تنها یک شیء را مشخص می کند در حالیکه اشیاء می توانند مختصات x, y, z و یا هر ترکیب دوتایی یکسان از آنها را داشته باشند. بنابراین مختصات سه بعدی اشیاء می تواند به عنوان یک مجموعه کلید برای جستجوی آنها استفاده شود.

در فصل گذشته استفاده از منحنی هیلبرت را به عنوان یکی از راه کارهای این مسأله مطرح نمودیم. در این فصل ساختار دیگری را بر مبنای منحنی ترتیب Z^1 و درخت KD مطرح خواهیم نمود که برای مسیریابی پیام های به روز رسانی به مختصات سه بعدی نقاط مناسب است.

استفاده از منحنی ترتیب Z که از خانواده منحنی های پرکننده فضا است، یک انتخاب مناسب برای اندیس دهی اطلاعات چند بعدی به شمار می رود. منحنی پرکننده فضای Z نیز مشابه با منحنی هیلبرت از نزدیکی تمام نقاط در فضا عبور می کند و به هر نقطه در فضای سه بعدی اندیس یک بعدی را نسبت می دهد که شماره ترتیب آن روی نقاط منحنی است. انگیزه اصلی استفاده از منحنی Z مشابه با منحنی هیلبرت، خواص مناسب آن در حفظ نسبت فاصله نقاط هنگام نگاشت آنها است. این ویژگی خصوصاً هنگام ارسال درخواست ها به نواحی پیوسته، کارایی جستجو را بسیار بالا می برد. در کاربرد محیط مجازی، منحنی Z هر نقطه در محیط مجازی را به شماره ترتیب نزدیک ترین نقطه منحنی Z به آن می نگارد.

به منظور مسیر یابی کارآمد پیام ها پس از نگاشت موقعیت نقاط بر روی منحنی ترتیب Z ، به

¹ Z-order curve

یک الگوریتم جستجوی مناسب در آن نیاز است که در این قسمت ما جستجو به روش درخت KD را پیشنهاد می کنیم. یکی از مهمترین انگیزه ها در استفاده از درخت KD کارایی آن در یافتن پاسخ به درخواست های بازه ای^۱ و به عبارت دیگر ارسال بهینه پیام ها به نواحی پیوسته در محیط مجازی است. بنابراین در ساختار پیشنهادی تعداد گام های طی شده در مسیر پیام های به روز رسانی به ویژه در سناریوهای ارسال به یک ناحیه به جای یک تک نقطه، تا حدود زیادی کاهش می یابد.

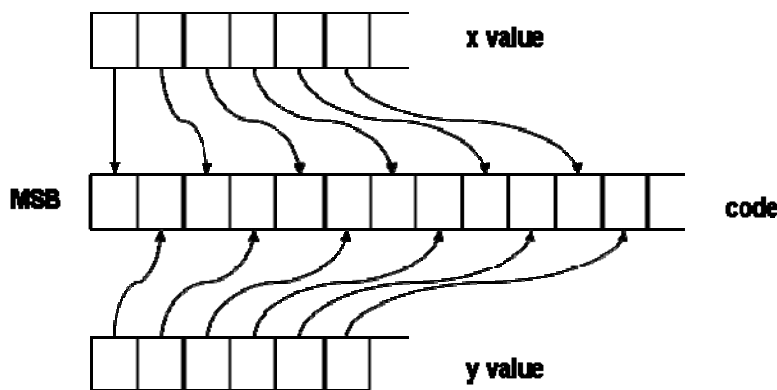
در زیر فصل های بعدی به تفصیل در مورد منحنی Z و مسیر یابی بر مبنای جستجو بر روی درخت KD بحث خواهیم نمود.

۱.۶. منحنی پرکننده فضای ترتیب Z

منحنی پرکننده فضای Z در سال ۱۹۹۶ توسط G.M. Morton پیشنهاد داده شد. منحنی ترتیب Z به علت خواص بسیار مناسب حفظ نزدیکی معمولاً در ساختارهای داده برای نگاشت اطلاعات چند بعدی به یک بعد مورد استفاده قرار می گیرد. منحنی Z برای نگاشت نقاط فضای چند بعدی به یک فضای فشرده، از یک نقطه فضا شروع می کند و فضا را به شکل پیوسته جاروب می کند به طوری که از نزدیکی تمامی نقاط عبور کند.

کلید یک نقطه در منحنی Z با روشی محاسبه می شود که در اصطلاح به آن “bitinterleaving” گفته می شود. در این روش در مرحله های متعدد بیت های متناظر از مختصات y, x و z نقاط برداشته می شود و در کنار هم قرار داده شده و خروجی آن در کنار خروجی مرحله قبل قرار داده می شود تا یک خروجی جدید ساخته شود [85][86]. این روند در شکل ۱-۶ نشان داده شده است.

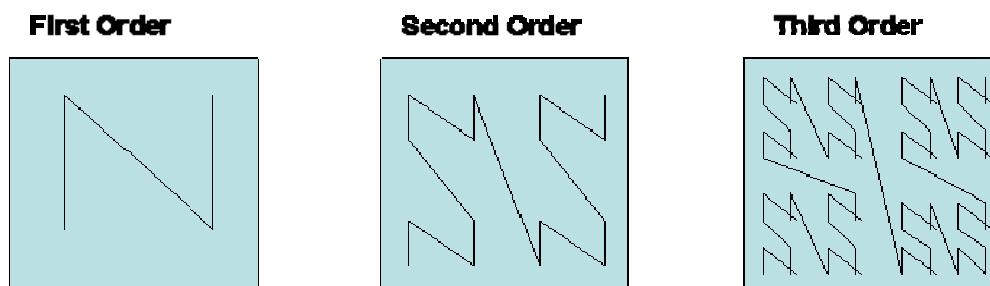
^۱ Range query



شکل ۱-۶ نگاشت Z

منحنی مرتبه اول Z به شکل یک Z چرخش یافته است و مرتبه های بالاتر آن با چرخش و انتقال منحنی مرتبه اول به دست می آید. منحنی های نشان داده شده در شکل ۲-۶ منحنی Z را در دو بعد و سه مرتبه نشان می دهد. با وجود اینکه کاربرد ما سه بعدی است، برای سهولت نمایش شکل ها در دو بعد آورده شده اند. ولی نکته قابل توجه آن است که مسایل شرح داده شده در دو بعد به آسانی به سه بعد قابل تعمیم است. همانطور که قبلا نیز اشاره نمودیم ویژگی مهم نگاشت Z که از کنار هم قرار دادن بیت های هم مرتبه حاصل می شود، حفظ نزدیکی نقاط هنگام نگاشت آنها از چند بعد به یک بعد است. به عبارت دیگر منحنی Z دسته بندی نقاط را هنگام نگاشت آنها حفظ می کند که این ویژگی در جستجو در بازه های فشرده می تواند بسیار کارآمد باشد.

در زیر فصل بعدی درخت KD را به عنوان ابزاری مناسب برای جستجو میان این اطلاعات



شکل ۲-۶ منحنی پرکننده فضای Z

دسته بندی شده ، معرفی خواهیم کرد. در اینجا تنها به این نکته مهم اشاره می کنیم که درخت KD در واقع ساختار درختی نمایش تبدیل Z خواهد بود که حرکت در امتداد شاخه های آن مسیر جستجو میان اطلاعات را تسهیل می کند.

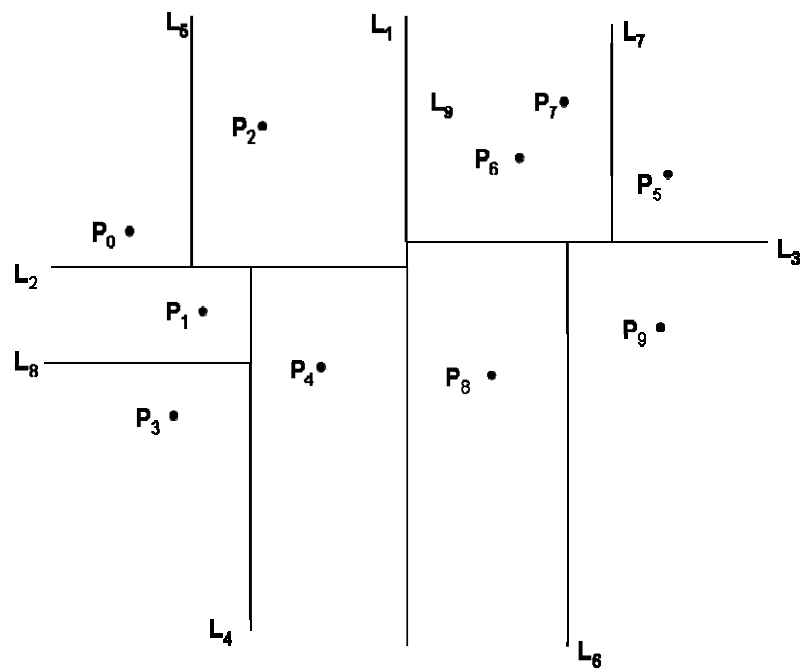
۲.۶. درخت های جستجوی چند بعدی دودویی (درخت KD)

درخت جستجوی چند بعدی دودویی یا درخت KD که حالتی خاص از درخت تقسیم فضا^۱ است [14] ، ساختار داده ای برای دست بندی نقاط در فضای چند بعدی است. درخت KD در واقع فرم توسعه یافته ای از درخت جستجوی دودویی در k بعد است و برای جستجوی یک رکورد با در دست داشتن k کلید در یک پایگاه داده به کار می آید. به طور کلی درخت جستجوی KD برای هر کاربردی که جستجو در میان اطلاعات چند بعدی را در بر داشته باشد، می تواند مفید واقع شود. تفاوت میان یک درخت جستجوی باینری و درخت KD در این است که در درخت جستجوی باینری استاندارد، مکان فیزیکی رکورد ها بر مبنای یک کلید مشخص می شود در حالیکه در درخت KD این مکان با تعدادی کلید مشخص می شود. به طور مثال اگر فرض کنیم که هر رکورد تعداد K کلید $K_1, K_2, \dots, K_n$ منحصر به خود دارد، در حرکت از ریشه درخت KD در جستجو برای یک رکورد، تصمیم حرکت به شاخه چپ یا راست درخت با مقایسه اولین کلید رکورد (K_1) با مقدار ذخیره شده در ریشه درخت صورت می گیرد. به همین ترتیب در انتخاب شاخه مرتبه دوم ، دومین کلید رکورد (K_2) با مقدار نقطه داخلی درخت مقایسه می شود و این روند تا مرتبه $n+1$ در درخت ادامه پیدا می کند که پس از این مرحله دوباره از اولین کلید در تصمیم گیری استفاده خواهد شد و روند قبلی تکرار می شود تا به برگ مناسب در درخت که متناظر با رکورد مورد جستجو است برسد.

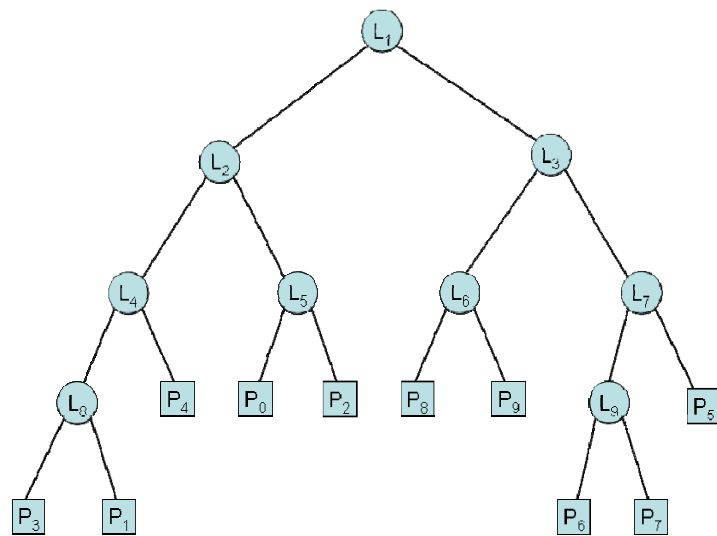
^۱ Binary Space Partitioning (BSP) Tree

فرآیند شرح داده شده در بالا می تواند برای جستجوی نقاط در فضای K بعدی با در دست داشتن مختصات آنها به کار رود. در این روند هر نقطه در فضا به عنوان یک رکورد از یک پایگاه داده در نظر گرفته می شود که مختصات آن همان کلیدهای مشخص کننده رکورد متناظرش هستند. برای مثال در فضای سه بعدی هر نقطه P رکوردی در نظر گرفته می شود که با سه مختصات x, y, z مشخص می شود. متناظر با آن درخت KD فضای سه بعدی را به زیر فضاهای متعدد تقسیم می کند.

فرآیند تقسیم بندی فضا و ساخت درخت KD در دو بعد برای یک توزیع نمونه غیر یکنواخت از نقاط در شکل ۳-۶ و ۴-۶ نشان داده شده است. همانطور که در شکل ۳-۶ دیده می شود، صفحه به شکل متواتر بر حسب مختصات x و سپس بر حسب مختصات y تقسیم بندی می شود به طوری که در هر تقسیم بندی تعداد نقاط موجود در هر دو طرف خط تقسیم با اختلاف حداکثر یک واحد با هم مساوی باشند. این تقسیم بندی تا جایی ادامه پیدا می کند که تعداد نقاط در هر زیر شاخه از عدد m که در این مثال برابر یک است، کمتر باشد. هنگام ساخت درخت KD متناظر با شکل ۳-۶، هر نقطه روی درخت ساخته شده متناظر با یک ناحیه مستطیلی در شکل متناظر می شود. به طور مثال ریشه درخت KD متناظر با تمام صفحه است و نقطه موجود در زیر شاخه سمت چپی آن متناظر با زیر صفحه سمت چپ آن است. روند ساخت درخت KD در تابع `Build_KDTree` به شکل شبه کد در شکل ۵-۶ نشان داده شده است. این تابع چهار ورودی دارد: ورودی P نشان دهنده مجموعه نقاط است، ورودی $depth$ عمق درخت KD ساخته شده را نشان می دهد، ورودی m مشخص کننده حداکثر نقاط قابل ذخیره در هر برگ درخت ساخته شده و به عبارت دیگر حداکثر نقاط موجود در هر واحد تقسیم نهایی است و در نهایت `Bbox` مستطیل احاطه کننده نقاط است..



شکل ۳-۶ تقسیم بندی فضا در دو بعد برای توزیع غیر یکنواخت نقاط



شکل ۴-۶ درخت KD متناظر

```

Build_KDTree (P, depth, Bbox, m) {
  if P contains m or fewer points then
    return a leaf node v storing these points and the node rectangle Bbox
  else if depth is even then
    Split P into two subsets with a vertical line l through
      the median x-coordinate of the points in P.
    Let P1 be the set of points to the left of or on l,
    and let P2 be the set of points to the right of l.
    Let Bbox1 be the part of Bbox to the left of l,
    and let Bbox2 be the part of Bbox to the right of l.
  else
    Split P into two subsets with a horizontal line l through
      the median y-coordinate of the points in P.
    Let P1 be the set of points below or on l,
    and let P2 be the set of points above l.
    Let Bbox1 be the part of Bbox below l,
    and let Bbox2 be the part of Bbox above l.
  vleft ← Build_KDTree (P1, depth + 1, Bbox1, m)
  vright ← Build_KDTree (P2, depth + 1, Bbox2, m)
  Create a node v storing l and the node rectangle Bbox,
  make vleft the left child of v, and
  make vright the right child of v.
  return v

```

شکل ۵-۶ شبه کد تابع تشکیل درخت KD

درخت KD امکان جستجوهای دقیق، تقریبی، ناحیه ای و بهترین تطبیق را فراهم می آورد و قابلیت های مناسبی در پشتیبانی همه این درخواست ها دارد. درخواست های دقیق^۱ ساده ترین نوع درخواست ها بر روی درخت KD هستند. این جستجو از ریشه درخت آغاز می شود و با انجام مقایسه

^۱ Exact Query

های مناسب مرحله به مرحله در امتداد مقصد مورد نظر روی درخت که پاسخ جستجو است حرکت می کند. چنین جستجویی نیاز به $O(\log(N))$ مقایسه دارد.

در درخواست های تقریبی^۱ به جای مشخص شدن تمام کلید ها تنها تعدادی از کلید های هر رکورد مشخص می شود و بنابراین پاسخ به یک درخواست تقریبی می تواند شامل چندین رکورد باشد. در الگوریتم مسیر یابی به نقاط محیط مجازی این درخواست ها متناظر با ارسال پیام های به روز رسانی به یک خط یا یک صفحه هستند. در جستجوی پاسخ به یک درخواست تقریبی در برخی شاخه ها که کلید متناظر برای تصمیم گیری مشخص نباشد، به جای حرکت در امتداد یک زیر شاخه، هر دو زیر شاخه انتخاب می شوند.

یکی دیگر از خانواده درخواست ها روی درخت KD، درخواست های ناحیه ای هستند که در آنها به جای مشخص شدن کلیدها به صورت یک تک مقدار، بازه ای از مقادیر برای هر کلید مشخص می شود. در هنگام اجرای این درخواست ها با شروع از ریشه درخت در هر شاخه حرکت ممکن است در مسیر یک یا هر دو زیر شاخه انجام شود که حرکت در مسیر هر دو شاخه زمانی رخ می دهد که آستانه تصمیم گیری داخل بازه مشخص شده برای کلید قرار بگیرد. در کاربرد محیط های مجازی این نوع از درخواست ها می تواند برای ارسال پیام های به روز رسانی به هر ناحیه ای در فضا به کار رود. در نهایت آخرین خانواده درخواست ها روی درخت KD درخواست های بهترین تطبیق^۲ هستند که در شرایطی به کار می روند که نیاز است در صورت عدم وجود نقطه مورد نظر درخواست به نزدیکترین همسایه آن مسیر یابی شود. در زیر فصل های بعدی در مورد چگونگی استفاده از این درخواست ها در مسیر یابی بحث خواهیم نمود.

^۱ Partial Match Query

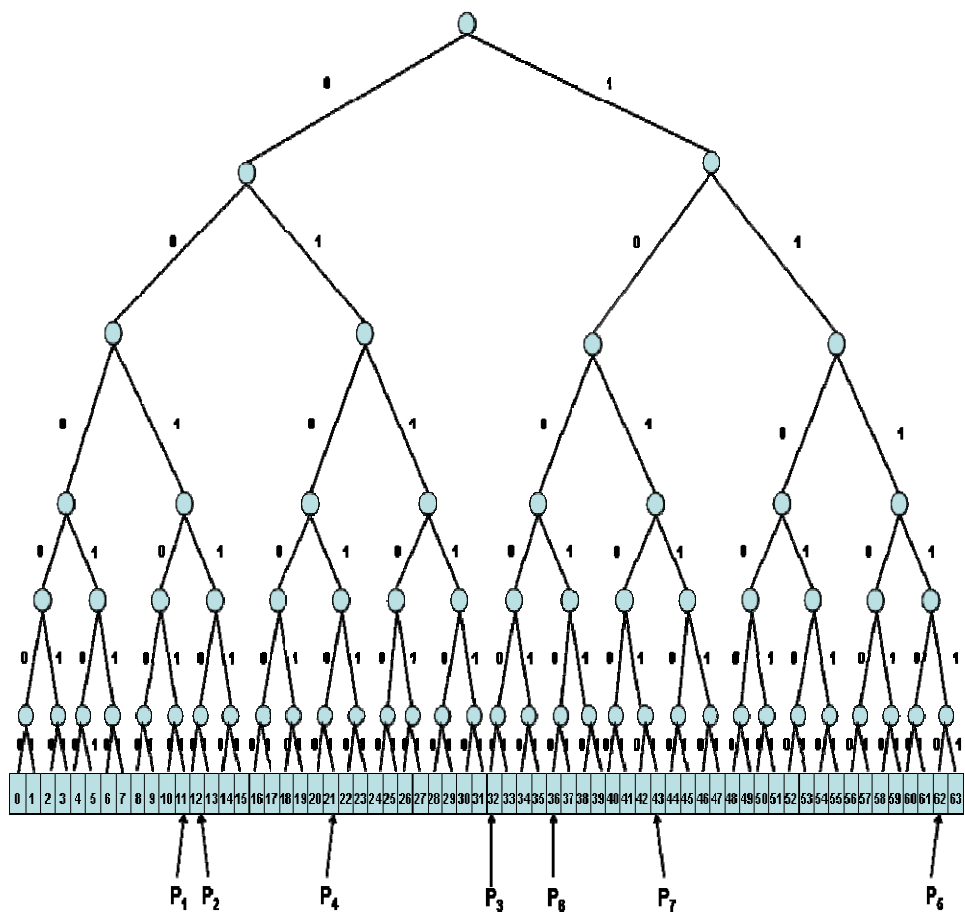
^۲ Best Match Query

۳.۶. نمایش منحنی Z در قالب درخت KD

همانطور که قبلاً اشاره کردیم، نگاشت Z برای نقاط فضای سه بعدی از کنار هم قرار دادن متناوب بیت های مختصات های x, y, z و به دست می آید. در درخت KD متناظر با منحنی z ، توالی شماره ترتیب^۱ برگ ها مشابه توالی شماره ترتیب آنها روی منحنی Z است. برای حرکت بر روی این درخت به منظور رسیدن به نقطه ای با یک شماره ترتیب خاص، حرکت از ریشه درخت انجام می شود و تصمیم حرکت به شاخه چپ یا راست با توجه به صفر یا یک بودن MSB عدد مورد نظر اتخاذ می شود و در شاخه های پایین تر بیت های مرتبه پایین تر برای تصمیم گیری مورد استفاده قرار می گیرند.

شکل ۶-۶ درخت KD متناظر با منحنی Z را به همراه تعدادی نقاط بر روی آن نشان می دهد. به طور مثال نقطه P1 با مختصات (۳و۱و۰) که معادل اعداد باینری (۱۱و۱و۰) را در نظر بگیرید که نگاشت آن بر روی منحنی Z بر طبق قوانین ذکر شده (۱۱و۱و۰) خواهد بود که معادل عدد دهدهی ۱۱ است. رشته (۱۱و۱و۰) مسیر حرکت روی شاخه های درخت را در شکل ۶-۶ برای رسیدن به P1 نشان می دهد. نقاط P1-P6 نمونه ای از نگاشت نقاط دیگر یک محیط مجازی با مختصات دیگر بر روی منحنی Z مرتبه ۲ هستند. نکته قابل توجه این است که اگر تعداد نقاط محیط مجازی بیشتر باشد، در نگاشت معمولاً به منحنی های مرتبه بالاتر نیاز خواهد بود. به طور مثال برای نگاشت مختصات ۱۲ بیتی نقاط در سه بعد به درختی با ۳۶ مرتبه نیاز است. با توجه به نمایش درختی شرح داده شده، مسیر یابی بسته ها به نقاط محیط مجازی هدایت آنها در مسیر مناسب بر روی شاخه های درخت برای رسیدن به مقصد مورد نظر با عبور از کمترین تعداد نقاط میانی است. در زیر فصل بعدی جزئیات الگوریتم مسیر یابی را شرح خواهیم داد.

^۱ Sequence Number



شکل ۶-۶ مثالی از درخت KD متناظر با منحنی Z

۴.۶. الگوریتم مسیر یابی

برای هدایت بسته ها در مسیر مناسب بر روی شاخه های درخت KD، هر نقطه باید به تعداد کافی همسایه برای خود نگهداری کند که تعداد آنها برای یک محیط مجازی سه بعدی با مختصات b بیت (متناظر با منحنی Z مرتبه b)، $12 * b$ خواهد بود.

جدول مسیر یابی هر نقطه چهار ستون و $3 * b$ سطر دارد که عضو موجود در سطر k ام، $k-1$ شاخه مشترک با نقطه مورد نظر دارد. به طور مثال برای تشکیل جدول مسیر یابی نقطه P_1 ، با شروع از ریشه درخت و حرکت به سمت نقطه P_1 ، در هر مرحله با حرکت در امتداد یک زیر شاخه، چهار نماینده از زیر شاخه دیگر به عنوان همسایه P_1 انتخاب می شوند که این چهار نماینده هر کدام از یک

زیر شاخه مستقل از آن شاخه انتخاب می شوند. جدول مسیر یابی هر نقطه، علاوه بر همسایگان آن نقطه شامل مجموعه ای از نزدیکترین نقاط به آن نقطه نیز می باشد که در اصطلاح به آن مجموعه برگ ها گفته می شود. همانطور که قبلا نیز اشاره نمودیم ، مجموعه برگ ها به همگرایی الگوریتم مسیر یابی در مراحل آخر کمک می کند.

از آنجا که سطر های آخر جدول مسیر یابی به علت محدود بودن کاندیداهای مناسب برای آنها و محدود بودن پیام های جستجوی همسایگان ممکن است خالی بمانند، در مراحل آخر مسیر یابی امکان آن وجود دارد که همسایه مورد نظر برای ارسال بسته در جدول مسیر یابی وجود نداشته باشد. بنابراین در چنین شرایطی وجود مجموعه برگ ها به نقطه ارسال کننده بسته کمک می کند تا بسته را به یک همسایگی نزدیک ارسال نماید تا در نهایت بسته مقصد مورد نظرش را پیدا کند. شکل ۶-۷ نمونه ای از جدول مسیر یابی را برای نقطه P_1 با عدد شناسه ۱۱ نشان می دهد که در آن خانه های خاکستری نشان دهنده همسایه هایی هستند که پیدا نشده اند.

Routing Table for object with key 11

Routing Set

Branch	0	1	2	3
Order				
1	32	43	52	62
2	18	22	26	28
3	0	3	4	6
4		13		15
5	8		--	--
6		--	--	--

Leaf Set

13	15	16
8	6	4

شکل ۷-۶ جدول مسیر یابی ساختار درخت KD

شکل ۸-۶ شبه کد الگوریتم مسیر یابی یک بسته به مقصد D را توسط نقطه X نشان می

دهد. در این شبه کد علائم اختصاری زیر به کار رفته است :

$R(j)$: عضو موجود در سطر j ام جدول مسیر یابی

$shl(A,B)$: طول پیشوند مشترک میان بیت های A و B

K : تعداد سطرهای جدول مسیر یابی

```

current_level <= shl(D,X)
M <= D(k-current_level..min(k-current_level+1,k))
If (current_level=K-1) then
    Route to (X) // the packet is destined for the node X itself
Else
    If ((R(current_level+1,M))!=null) then
        Route to(R(current_level+1,M))
    Else
        Route to (L) st. L ∈ Leaf_set ∪ X, |L-D| : min
    End If
End If

```

شکل ۸-۶ شبه کد الگوریتم مسیر یابی بر مبنای درخت KD

همانطور که قبلاً هم اشاره نمودیم ، ارسال پیام های به روز رسانی به نواحی پیوسته در محیط مجازی کاربرد بسیاری دارد. چند پخششی پیام ها به یک ناحیه پیوسته در فضا مشابه انجام درخواست های جستجوی ناحیه ای بر روی درخت KD است. در این نوع ارسال ها به جای مشخص شدن دقیق x ، y و z نقطه مقصد پیشوندی برای این مختصات مشخص می شود. به طور مثال اگر مختصات x را ۲ ، y و z بیتی در نظر بگیریم پیشوند باینری ۱۱۰ ناحیه با x های حوزه ۱۰ تا ۱۱ ، y های ۱۰ تا ۱۱ و z های ۰۰ تا ۰۱ را مشخص می کند.

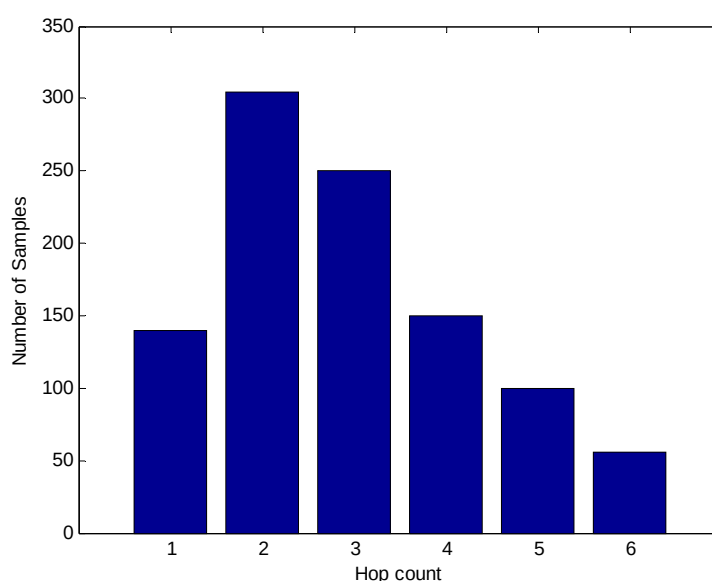
الگوریتم مسیر یابی هنگام تصمیم گیری در مورد ارسال بسته ای که پیشوند مقصد آن مشخص شده است، این بسته ها را به نقاط هدایت می کند که پیشوند آنها با پیشوند مقصد تطابق دارد. این شرط در رابطه (۱-۶) نشان داده شده است.

$$\text{Route to } (Y) \text{ st. } Y \in \text{Leaf_set} \cup X \cup R, \text{ shl}(Y, \text{prefix}) : \max \quad (1-6)$$

۵.۶. نتایج شبیه سازی

در این قسمت ما نتایج شبیه سازی الگوریتم مسیریابی پیشنهادی بر مبنای منحنی هیلبرت را ارائه خواهیم داد. شبیه سازی ها در یک محیط مجازی گسترده با ۱۰۰۰۰ نقطه و در شرایط تعریف شده در فصل ۳ انجام شده است. برای تخصیص شناسه به کاربران از منحنی Z مرتبه ۱۰ در سه بعد استفاده شده است. این منحنی معادل با یک درخت KD از مرتبه ۳۰ خواهد بود.

شکل ۹-۶ نتایج شبیه سازی تعداد گام ها را در شرایط ذکر شده نشان می دهد. در این شکل متوسط تعداد گام ها در ارسال بسته ها ۲.۹ است و بیشترین تعداد گام ۶ است. در حالیکه بیشتر پیام ها در کمتر از ۴ گام به مقصد رسیده اند. شکل هیستوگرام این قسمت شباهت های زیادی به هیستوگرام به دست آمده از منحنی هیلبرت دارد که این شباهت به دلیل ساختار مشابه در شبکه و ترافیک و علاوه بر آن شباهت های ذاتی در مبنای عملکرد این دو ساختار است. مقایسه این شکل با شکل مشابه در فصل قبل نشان می دهد که عملکرد ساختار هیلبرت در مقایسه با ساختار درخت KD اندکی بهتر است.

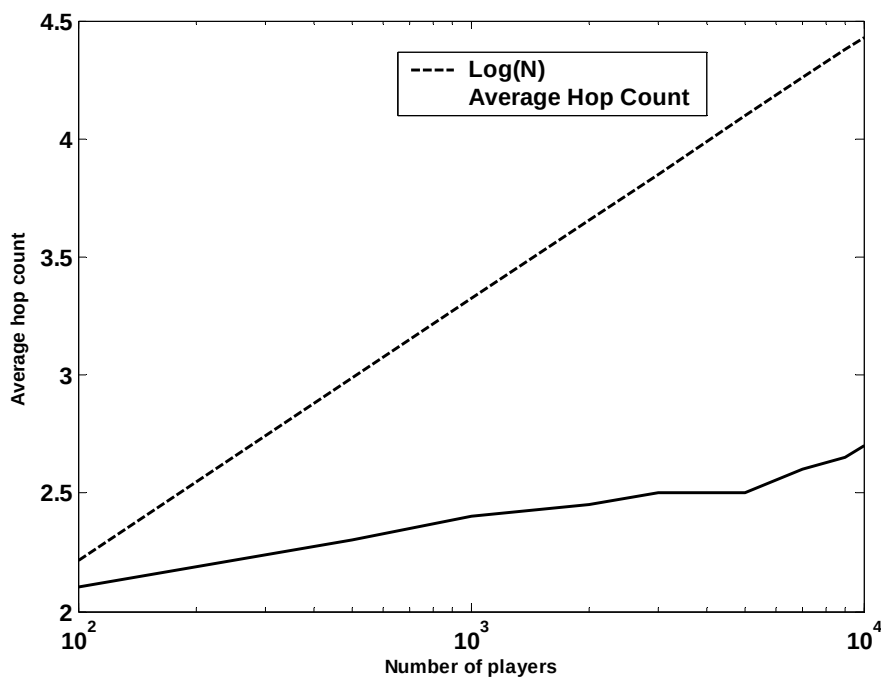


شکل ۹-۶ هیستوگرام گام های ارسال بسته ها

۱.۵.۶. قابلیت توسعه

در این قسمت ارتباط میان گام های مسیر یابی را با تعداد کاربران محیط مجازی بررسی خواهیم نمود. به این منظور تعداد کاربران محیط مجازی از ۱۰۰ تا ۱۰۰۰۰ تغییر داده شده است و ابعاد محیط مجازی نیز به نسبت آن تنظیم شده است به طوری که میانگین تعداد کاربران در واحد سطح ثابت بماند. مرتبه منحنی هیلبرت برای حالت ۱۰۰۰۰ کاربر مساوی ۱۰ قرار داده شده است و در سایر موارد به نسبت لگاریتم تعداد کاربران مقدار کمتری برای آن انتخاب شده است. در هر مرحله از شبیه سازی مقدار متوسط تعداد گام های مسیر در شبکه محاسبه و ثبت شده است. شکل ۶-۱۰ متوسط تعداد گام ها را در هر مرحله در مقابل تعداد نقاط شبکه در آن مرحله نشان می دهد.

همانطور که در نتایج شبیه سازی دیده می شود، ساختار پیشنهادی تعداد گام های ارسال بسته ها



شکل ۶-۱۰ متوسط تعداد گام های مسیر بسته ها بر حسب تعداد نقاط شبکه

را در حد قابل قبولی کوچک نگه می دارد. نکته قابل توجه آن است که الگوریتم های جستجوی همسایه های نزدیک و مسیرهایی نزدیک می توانند مشابه فصل قبل و با تغییرات جزئی در کنار ساختار مبتنی بر منحنی هیلبرت مورد استفاده قرار بگیرند. به دلیل شباهت زیاد این الگوریتم ها به الگوریتم های ذکر شده در فصل قبل، ما از تکرار آنها در اینجا پرهیز می کنیم و تنها نتایج شبیه سازی مربوط به تعداد گام ها را ارائه می دهیم در حالیکه کمینه سازی طول گام ها از لحاظ تأخیر با روش های ذکر شده قابل انجام است.

فصل هفتم

ساختار مبتنی بر مسیریابی هندسی

اصول LSH

نگاشت سلسله مراتبی

نگاشت با چند جدول

الگوریتم مسیریابی

مدیریت تغییرات در شبکه

نتایج شبیه سازی

همانطور که در فصول پیشین نیز اشاره کردیم، ارسال پیام های به روز رسانی به مختصات های مورد نظر در محیط مجازی به جای ارسال پیام ها به آدرس IP نقاط موجود در این مختصات ها نیاز به ارسال درخواست های متعدد برای یافتن موقعیت نقاط را برطرف می کند و به طراحی یک محیط مجازی نظیر به نظیر با ساختار گسترده کمک می کند.

برای ارسال پیام های به روز رسانی به مکان های مورد نظر در محیط مجازی به یک سرویس مسیریابی نیاز است که بتواند بسته ها را نقاط مشخص در محیط مجازی ارسال نماید. بنابراین استفاده از مسیریابی هندسی در این شرایط انتخابی مناسب به نظر می رسد. مسیریابی هندسی که به مسیریابی جغرافیایی^۱ یا مسیریابی مکانی^۲ نیز معروف است، از خانواده مسیریابی های بدون حافظه^۳ است. در این مسیریابی، بسته ها در چند گام به موقعیت مشخص شده در فضای مسیریابی منتقل می شوند به طوری

^۱ Geographical Routing

^۲ Location based Routing

^۳ In_line

که در هر گام بسته مقداری به مقصد نزدیک تر شود. در مسیریابی هندسی بسته در تمام گام های مسیر باید به مقصد نزدیک تر شود، بنابراین بسته های منتقل شده در یک همسایگی محلی از طریق نقاط دور از همسایگی منتقل نخواهند شد. ویژگی مسیریابی محلی بسته ها در ساختار مسیریابی هندسی یکی از مهمترین ویژگی های مطلوب استفاده از آن است. این ویژگی به کاهش تأخیر ارسال بسته ها کمک می کند.

مسیریابی هندسی که در حوزه شبکه های بیسیم تعریف شده است، برای به کار گیری در کاربرد محیط های مجازی نیاز به تغییرات ساختاری متعددی دارند. در این فصل ساختار پیشنهادی برای محیط های مجازی بر مبنای مسیریابی هندسی ارائه خواهد شد که در این ساختار ویژگی های متعددی به ساختار پایه مسیریابی هندسی اضافه می شود. مهمترین این ویژگی ها استفاده از مسیریابی سلسله مراتبی برای کاهش گام های مسیر ارسال و استفاده از چندین شبکه مستقل همزمان به عنوان مرجع مسیر یابی است که در ادامه جزئیات آن را شرح خواهیم داد.

ایده استفاده از اطلاعات موقعیت نقاط هنگام مسیریابی نخستین بار در سال ۱۹۸۰ و در حوزه شبکه های رادیویی بسته ای^۱ مطرح شده است [90][89]. مسیریابی هندسی در اصطلاح به خانواده مسیر یابی هایی گفته می شود که برای انتقال بسته ها از یک مبدا تا یک مقصد، از اطلاعات جغرافیایی و مکان مقصد به جای آدرس آن استفاده می کنند. مسیر یابی هندسی می تواند در شرایطی که بسته ها باید به موقعیت های مشخص در یک شبکه ارسال شوند و نقاط واسط انتقال اطلاعاتی از توپولوژی شبکه ندارند، مفید باشد. همچنین در شرایطی که جمع آوری اطلاعات توپولوژی شبکه به دلیل گسترده بودن ابعاد و یا تغییرات مداوم در آرایش شبکه، هزینه زیادی را به دنبال دارد، استفاده از مسیر یابی هندسی به

^۱ Packet based Radio Networks

ایجاد یک ساختار گسترده و توسعه پذیر کمک می کند.

در مسیریابی هندسی فرض بر آن است هر نقطه شبکه با داشتن اطلاعات زیر در مورد انتقال هر بسته دریافتی تصمیم گیری می کند :

- مختصات (x,y,z) که خودش در آن قرار دارد
 - مختصات تعداد محدودی نقطه که همسایه های نقطه مورد نظر فرض می شوند.
 - مختصات نقطه ای که بسته در نهایت باید به آن برسد.
- با در دست داشتن اطلاعات بالا، هر نقطه می تواند در مورد بهترین انتخاب از میان همسایگانش بهترین آنها را برای انتقال بسته دریافتی انتخاب کند. برای هر نقطه مقصد، به زیر مجموعه ای از همسایگان که به آن مقصد خاص نزدیک تر از نقطه فعلی هستند در اصطلاح مجموعه انتقال^۱ برای آن مقصد گفته می شوند. نقطه انتقال دهنده بسته بر مبنای معیارهای از پیش تعیین شده عضوی را از بین مجموعه انتقال برای ارسال بسته انتخاب می کند. در مسیریابی هندسی مرتبه همسایه های هر نقطه و پیام های جابجا شده برای کنترل توپولوژی $O(1)$ است و با گسترش ابعاد شبکه افزایش نمی یابد.

اتخاذ تصمیم مناسب در انتقال بسته ها موضوع تحقیقات گسترده ای در موضوع مسیریابی هندسی بوده است و روش های متعددی در حوزه شبکه های بیسیم پیشنهاد شده است. روش انتقال حریص نقطه ای را در مجموعه انتقال برای ارسال بسته انتخاب می کند که کمترین فاصله را تا مقصد بسته داشته باشد. مسیریابی قطب نما^۲ [91] نیز یک الگوریتم مسیریابی بدون حافظه است که در آن هر نقطه یک بسته را در میان همسایگان خود را به همسایه ای ارسال می کند که خط اتصال با آن

^۱ Forwarding Set

^۲ Compass Routing

کوچکترین زاویه را با خط اتصال با مقصد بسته می سازد. مسیریابی تصادفی قطب نما^۱ [92] نوع دیگری از مسیریابی قطب نمایی است که در آن گام بعدی برای ارسال بسته به صورت تصادفی از میان همسایه هایی که کوچکترین زاویه را در بالا و پایین خط متصل کننده مبدا با مقصد می سازند، انتخاب می شود. در روش MFR^۲ [93]، نقاط بسته ها را به همسایه هایی منتقل می کنند که بیشترین پیشرفت را در جهت محور x برای رسیدن به مقصد داشته باشد. GEDIR^۳ [94] نیز الگوریتم مسیریابی دیگری از خانواده الگوریتم های حریص است که در آن بسته بدون توجه به فاصله نقطه انتقال دهنده از مقصد، به نزدیک ترین همسایه به مقصد ارسال می شود.

نکته قابل توجه آن است که هیچ یک از الگوریتم های ذکر شده همگرایی مسیریابی را گارانتی نمی کنند. این الگوریتم ها در شرایطی که بسته ها به می نیم های محلی یا حفره ها می رسند، قادر به ارسال بسته به مقصد اصلیش نیستند. در یک نقطه می نیم محلی برای یک بسته، هیچ همسایه دیگری نزدیک تر از نقطه مورد نظر به مقصد بسته پیدا نمی شود و به عبارت دیگر مجموعه انتقال آن نقطه برای مقصد خاصی تهی باشد. تعداد بسیاری از این الگوریتم های مسیریابی در شرایطی که توپولوژی شبکه ویژگی DT^۴ را داشته باشد، انتقال بسته ها را گارانتی می کنند [92]. ولی ساخت و به روز رسانی چنین توپولوژی هزینه زیادی را به همراه دارد. در ساختارهای مبتنی بر توپولوژی های عمومی اجتناب از حفره ها با به کار گیری الگوریتم هایی مانند Face-1 یا Face-2 [95] امکان پذیر است. ولی استفاده از این روش ها کارایی الگوریتم مسیریابی را تا حد زیادی کاهش می دهد. برای حل این مشکل روش

^۱ Randomized Compass Routing

^۲ Most Forwarding within Radius

^۳ GEographical DIstance Routing

^۴ Delaunay Triangulation

هایی مانند¹ GPSR [95]،² GFG [96]،³ GFPG [97] و⁴ GOAFR+ [98] از ترکیبی از روش های حریص و مسیریابی Face برای بهبود کارایی و حفظ همگرایی استفاده می کنند.

نکته قابل توجه توجه در مورد تحقیقات مرتبط آن است که بیشتر الگوریتم های مسیریابی ارائه شده بیشتر در شبکه های بیسیم بوده اند و تحقیقات معدودی در زمینه مسیریابی هندسی در شبکه های سیمی انجام شده است [99]. ساختار هندسی پیشنهادی در ویژگی های اساسی زیر با سایر الگوریتم های پیشنهادی در شبکه های بیسیم تفاوت دارد.

- تحقیقات حوزه محیط های مجازی بیشتر مسیریابی در دو بعد را مورد بررسی قرار داده اند و بسیاری از جنبه های مسیریابی سه بعدی مورد بررسی قرار نگرفته است و تحقیقات محدودی در این زمینه انجام شده است [100]. همچنین بیشتر تحقیقات انجام شده در سه بعد بیشتر بر مبنای نگاشت سه بعد در دو بعد می باشد.

- حفره ها در شبکه های بیسیم در نتیجه موانع و یا نواحی کم تراکم ایجاد می شوند که در این نواحی شعاع همسایگی نقاط برای پوشش تعداد کافی از همسایگان بزرگ نمی باشد. ولی محدودیت موانع در شبکه های سیمی وجود ندارد و در این شبکه ها حفره ها تنها در نقاط کم تراکم امکان رخ دادن دارند. بنابراین انتخاب درست همسایگی ها در این شبکه ها می تواند مشکل حفره ها را حل کند و نیازی به استفاده از الگوریتم های ذکر شده که مسیریابی را پیچیده تر کرده و کارایی آن را کمتر می کند، وجود ندارد.

- در شبکه های بیسیم، نقاط تنها امکان برقراری ارتباط مستقیم با همسایه های نزدیک خود را از

¹ Greedy Perimeter Stateless Routing

² Geographic-Forwarding-Geocast

³ Geographic-Forwarding-Perimeter-Geocast

⁴ Greedy and Other Adaptive Face Routing

طریق پیوندهای رادیویی با طول محدود دارند. چنین الگوریتمی به دلیل عدم وجود گام های بلند، هنگام مسیر یابی بسته ها را در همسایگی نزدیک خود حفظ می کند. ولی در عین حال عدم وجود گام های بلند باعث افزایش تعداد گام های مسیر یابی هنگام ارسال بسته ها به نقاط دور خواهد شد.

- حداکثر فاصله ارسال دستگاه های بیسیم به دلیل محدودیت در رادیوی آنها، به شعاع ثابتی در اطراف آنها محدود می باشد. بنابراین تعریف همسایگی ها از این محدودیت فیزیکی تأثیر می پذیرد. این محدودیت در تعریف همسایگی، امکان مسیر یابی سلسله مراتبی در شبکه را از بین می برد. چنین محدودیت فیزیکی در محدوده ارسال و تعریف همسایگی ها در شبکه های سیمی وجود ندارد. بنابراین امکان تعریف مسیر یابی هندسی سلسله مراتبی در آنها وجود دارد.
- محلی سازی پیام های به روز رسانی یک نقطه که معمولاً در ناحیه محدودی اطراف آن مبادله می شوند، می تواند در افزایش کارایی الگوریتم مسیر یابی مفید واقع شود. چنین شکل محلی ارسال ترافیک در الگوریتم های مرتبط پیشنهادی مفروض نبوده است.
- در شبکه های بیسیم نقاط به آسانی امکان انجام همه پخش را به نقاط اطراف خود دارند. به علت دسترسی به این کانال مناسب همه پخش، پیدا کردن همسایه ها را در این شبکه ها بسیار ساده تر از شبکه های سیمی است. در شبکه های سیمی عمل چند پخش در لایه فیزیکی انجام نمی شود و باید در لایه های بالاتر انجام پذیرد.

همانطور که در موارد بالا اشاره نمودیم، یکی از نکات مهم در طراحی پیشنهادی، مسیر یابی سلسله مراتبی در ساختار هندسی می باشد. در طرح پیشنهادی مسیر یابی سلسله مراتبی بر روی یک

افراز شطرنجی^۱ از فضا انجام می شود. مسیریابی بر روی یک ساختار شطرنجی تعریف همسایگی ها را آسان تر می کند و تحلیل همگرایی مسیریابی را نیز تسهیل می کند. ولی نکته قابل توجه آن است که نگهداری و به روز رسانی ساختار شطرنجی هزینه ای را به سیستم تحمیل خواهد کرد.

مسیریابی سلسله مراتبی روی ساختار شطرنجی نیاز به تقسیم سلسله مراتبی فضا به تعدادی زیرفضا دارد به طوریکه نقاط نزدیک به هم در زیر فضاهای یکسان قرار بگیرند. بدین منظور در ساختار پیشنهادی از تکنیک^۲ LSH [85] استفاده می شود. LSH در اصطلاح به مجموعه توابعی گفته می شود که نقاط یک فضای چند بعدی را به رشته های تک بعدی می نگارند. به عبارت دیگر LSH نقاط فضا را دسته بندی می کند و در این دسته بندی نقاط نزدیک به هم را در ظرف های یکسان قرار می دهد. بنابراین LSH در خانواده نگاشت های حافظ محلیت^۳ دسته بندی می شود.

در حوزه پایگاه های داده، LSH به عنوان ابزاری برای جستجوی سریع یک همسایگی در میان تعداد زیادی نقطه به شمار می رود. مسیریابی هندسی بسته ها به یک موقعیت خاص در شبکه مشابه جستجوی آن موقعیت در پایگاه داده شامل تمام نقاط محیط مجازی است. بنابراین LSH می تواند برای دسته بندی نقاط محیط مجازی در جهت جستجوی سریع میان آنها مفید واقع شود. در بخش های بعدی جزئیات طراحی ساختار پیشنهادی را شرح خواهیم داد.

۱.۷. اصول LSH

همانطور که قبلا نیز اشاره کردیم، LSH یکی از روش های معروف برای کاهش ابعاد اطلاعات چند بعدی هنگام جستجو میان آنهاست. ایده اصلی LSH تقسیم بندی نقاط یک فضای چند بعدی میان

^۱ Grid

^۲ Locality Sensitive Hashing

^۳ Locality Preserving

تعدادی ظرف است به طوریکه نقاط مشابه با احتمال زیادی در ظرف های مشابه قرار بگیرند.

توابع LSH گروهی از توابع $h \in H$ هستند که نقاط یک فضای d بعدی را به یک بعد می نگارند. به صورتی شروط زیر برقرار باشد :

- احتمال قرار گرفتن در یک ظرف برای هر دو نقطه p و q که فاصله آنها از حد معینی کمتر باشد، زیاد است.

$$PH[h(p) = h(q)] \geq P1 \text{ for } \|p - q\| \leq R1 \quad (1-7)$$

که در آن $R1 > 0$ آستانه مشخص شده توسط تابع LSH است.

- احتمال قرار گرفتن در یک ظرف برای هر دو نقطه p و q که دور از یکدیگر واقع شده اند، کم خواهد بود.

$$PH[h(p) = h(q)] \leq P2 \text{ for } \|p - q\| \geq cR1 \quad (2-7)$$

که در رابطه بالا $c > 1$ پارامتری است که در اصطلاح فاکتور تقریب تابع LSH خوانده می شود. مجموعه پارامترهای $(R1, cR1, P1, P2)$ حساسیت تابع LSH را مشخص می کنند.

جستجوی همسایه های نزدیک با کمک LSH در دو مرحله اصلی انجام می شود: در نخستین مرحله، خانواده G از توابع نگاشت تعریف می شود که هر تابع $g \in G$ متعلق به آن از به هم پیوستن k عدد از توابع $h_1, \dots, h_k$ که به صورت تصادفی از میان مجموعه H از توابع LSH انتخاب شده اند. هر تابع g_i یک نگاشت است که بر روی هر نقطه مانند P را بر طبق رابطه زیر عمل می کند :

$$g_i(p) = [h_1(p), \dots, h_k(p)] \quad i=1 \dots L \quad (3-7)$$

که در آن k و L به ترتیب عرض توابع نگاشت و تعداد آنها را مشخص می کنند.

با در دست داشتن یک نقطه مورد جستجو مانند q ، الگوریتم به تعداد L تابع g_i را برای نگاشت به کار می برد و نقاط را که در فرآیند نگاشت توسط هر یک از این توابع با q در یک ظرف قرار می گیرند، پیدا می کند. فرآیند جستجو زمانی متوقف می شود که نقطه ای با فاصله ای کمتر از یک میزان آستانه مشخص از q پیدا شود.

روش پیشنهادی ما فضای محیط مجازی را به تعدادی ناحیه تقسیم می کند که هر ناحیه در حکم یک ظرف نگاشت است. نگاشت نقاط محیط مجازی به این ظرف ها توسط خانواده ای از توابع نگاشت صورت می گیرد که بر مبنای بازه قرار گرفتن مختصات x, y و z نقاط در مورد ظرف مناسب برای نگاشت آنها تصمیم می گیرند.

$$h_i^x(p) = \lfloor (i+1)p_x \rfloor \quad (4-7)$$

$$h_i^y(p) = \lfloor (i+1)p_y \rfloor \quad (5-7)$$

$$h_i^z(p) = \lfloor (i+1)p_z \rfloor \quad (6-7)$$

$$g_i(p) = [h_i^x(p), h_i^y(p), h_i^z(p)] \quad (7-7)$$

بر طبق روابط بالا نگاشت نقاط به ظرف ها بر مبنای موقعیت فضایی آنها صورت می گیرد. به ازای مقادیر مختلف i توابع $g_i(p)$ فضا را به گونه های مختلف تقسیم می کنند. برای $i=1$ توابع نگاشت محورهای x, y, z را به دو نیمه تقسیم می کنند که در نتیجه آن فضای مکعبی محیط مجازی به ۸ مکعب کوچکتر تقسیم می شود.

نکته قابل توجه آن است که توابع ذکر شده نگاشتی از نوع یکنواخت انجام می دهند که بدون توجه به چگالی اوتارها و اشیاء در قسمت های مختلف محیط مجازی آن را به شکل یکنواخت افراز می کند. این در حالی است که یک مکانیزم نگاشت بهینه تر می تواند با دسترسی به الگوی تراکم در محیط

مجازی، ظرف های بیشتری را به نقاط پرتراکم اختصاص دهد و از اختصاص ظرف های غیر ضروری به نقاط خلوت اجتناب کند. در بقیه قسمت های این فصل ما از توابع یکنواخت تعریف شده در قسمت قبل استفاده خواهیم نمود، ولی این نکته را خاطر نشان می کنیم که تمامی راه حل های ارائه شده در حضور توابع غیر یکنواخت نگاشت کماکان قابل استفاده هستند.

۲.۷. نگاشت سلسله مراتبی

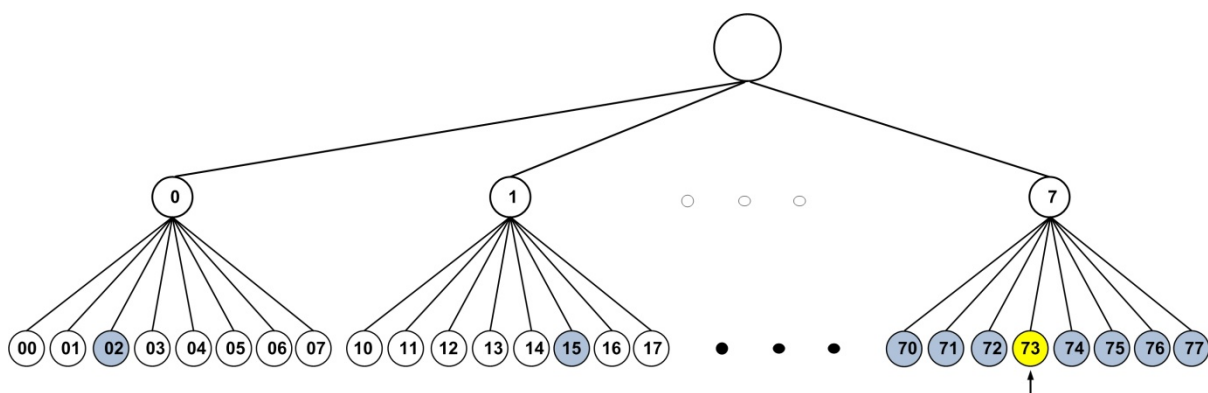
روش های مسیریابی پیشنهادی در محیط های مجازی دو مرحله اصلی را شامل می شوند: در نخستین مرحله جستجو برای ظرفی که نقطه مورد نظر در آن واقع شده است صورت می پذیرد و در مرحله بعدی جستجوی خطی میان نقاط ظرف پیدا شده انجام می شود تا نقطه مقصد پیدا شود. جستجوی خطی در میان نقاط زیاد معمولاً کارایی مناسبی ندارد. بنابراین تعداد نقاط درون ظرف ها باید تا حد امکان کم باشد. در نگاشت تعداد معینی نقطه این به معنای افزایش تعداد ظرف های فضای نگاشت است که سبب افزایش تعداد همسایه های هر نقطه و هزینه نگهداری جدول های مسیریابی خواهد شد. نگاشت سلسله مراتبی یک راه حل مناسب برای این مسأله است که در آن نقاط با وضوح متفاوت به مجموعه ظرف هایی نگاشته می شوند که رده های متناظر با وضوح بالاتر زیر مجموعه رده های متناظر با وضوح پایین تر هستند.

برای جستجوی یک نقطه ابتدا ظروف رده بالا جستجو می شوند و پس از پیدا شدن ظرف مورد نظر، جستجو درون ظروف رده پایین زیر مجموعه آن ادامه می یابد. بنابراین تعداد ظروف رده بالا بسیار کمتر از تعداد نقاط محیط مجازی است و این نسبت در مورد ظروف مرتبه پایین تر افزایش می یابد. بنابراین نگاشت سلسله مراتبی به ما این امکان را می دهد که در عین پایین نگاه داشتن هزینه مسیر یابی، فرآیند جستجو را تسریع نماییم.

شکل ۷-۱ نگاشت سلسله مراتبی فضای سه بعدی را توسط تابع g_1 ، توصیف شده در رابطه (۷-۷)، در دو مرتبه به شکل درختی نشان می دهد. نخستین رده در درخت متناظر با نخستین مرحله نگاشت است که در آن فضا به هشت زیر فضا تقسیم می شود که از ۰ تا ۷ شماره گذاری شده اند. دومین رده در درخت تقسیم هر زیر فضا را به هشت زیر فضای کوچکتر نشان می دهد که شماره آنها از ۰ تا ۷ به هم پیوستن شماره زیر فضای مادر آنها با اعداد ۰ تا ۷ به دست می آید. روش مشابهی می تواند برای توسعه نگاشت به مراتب پایین تر به کار گرفته شود.

۳.۷. نگاشت با چند جدول

با وجود آنکه مسیریابی سلسله مراتبی بر روی یک زیرساخت شطرنجی مزایای متعددی دارد، در برخی مواقع محدودیت حرکت بر روی ساختار شطرنجی سبب می شود که بسته ها در مسیریابی راه های طولانی تری را طی کنند. این اتفاق زمانی رخ می دهد که نقاط نزدیک به هم در محیط مجازی به ظرف های متفاوت نگاشته شوند. بنابراین نسبت فاصله نقاط هنگام حرکت روی ساختار شطرنجی حفظ نمی شود. راه حل پیشنهادی در برای حل این مسأله استفاده از چندین شبکه شطرنجی مستقل است به



شکل ۷-۱ فرآیند سلسله مراتبی نگاشت در دو مرتبه

طوری‌که هنگام طولانی شدن مسیر یابی روی یک شبکه، مسیر مورد نظر روی شبکه دیگر پیموده شود. برای حفظ ویژگی همگرایی مسیریابی بسته‌ها همیشه در مسیر یابی در عین جابجایی میان شبکه‌ها، قانون دیگری به مسیر یابی اضافه می‌شود. این قانون سبب می‌شود که بسته‌ها هنگام جابجایی میان دو شبکه شطرنجی همیشه در سلسله مراتب به سمت پایین حرکت کنند. در ساختار پیشنهادی، دو شبکه شطرنجی مستقل مورد استفاده قرار گرفته است که متناظر با توابع g_1 و g_2 بر طبق رابطه (۷-۷) هستند. شبکه نخست متناظر با تقسیم فضا به ۸ زیر فضا در هر مرحله است و شبکه دوم متناظر با تقسیم فضا به ۲۷ زیر فضا می‌باشد. از آنجا که عددهای ۲ و ۳ نسبت به هم اول هستند، هیچ مضرب مشترکی پیدا نخواهند کرد و بنابراین دو شبکه شطرنجی همیشه مرزهای مستقل خواهند داشت. بدین ترتیب اگر دو نقطه واقع در کنار مرزهای یک شبکه واقع باشند، در شبکه دیگر این دو نقطه در یک خانه قرار خواهند گرفت.

۴.۷. الگوریتم مسیریابی

با تقسیم محیط مجازی به مجموعه سلسله مراتبی از خانه‌ها، در این قسمت الگوریتم پیشنهادی برای مسیریابی بهینه بسته‌ها میان این خانه‌ها ارائه خواهد شد. با توجه به نمایش درختی نشان داده شده در شکل ۷-۱، مسیرهای مناسب انتقال پیام‌ها میان نقاط بر مبنای حرکت در امتداد شاخه‌های درخت تعریف می‌شوند. بنابراین هر نقطه میانی دریافت‌کننده یک پیام سعی می‌کند پیام را در مسیر شاخه مناسب برای رسیدن به مقصد هدایت نماید. مسیر سیستماتیک برای انتقال بسته‌ها بر روی درخت به صورتی است که بسته‌ها در هر گام یک مرتبه به مقصد نهایی خود نزدیک‌تر می‌شوند. به این منظور هر نقطه باید همسایه‌های کافی را در میان نقاط درخت برای خود نگهداری کند.

هر نقطه بر روی درخت برای انتخاب همسایه‌هایش از ریشه درخت به سمت موقعیتش در

پایین درخت حرکت می کند و در هنگام ورود به هر شاخه، یکی از برگ های زیرین شاخه های دیگر هم مرتبه آن را به عنوان همسایه برای خود انتخاب می کند. بدین ترتیب هر نقطه در هنگام دریافت یک بسته امکان هدایت آن را به زیر شاخه ای که یگ گام به مقصد نزدیک تر است، خواهد داشت.

به طور مثال در شکل ۷-۱ مجموعه همسایه ها برای نقطه ای در ظرف شماره ۷۳ که با علامت فلش مشخص شده است، نشان می دهد. نقطه شماره ۷۳ یک نماینده از ظرف های ۷۷-۷۰ را به عنوان همسایگان مرتبه اول خود انتخاب می کند. همچنین نقطه ۷۳، هفت همسایه دیگر را از شاخه های ۶-۰ برای خود انتخاب می کند. در شکل ۷-۱ تنها همسایه های انتخاب شده از شاخه های ۰ و ۱ نشان داده شده است که نقاط از ظرف های ۰۲ و ۱۵ انتخاب شده اند. این گروه، همسایگان مرتبه دوم نقطه ۷۳ به شمار می روند.

اگر درخت متناظر با شبکه، سلسله مراتب بیشتری داشته باشد، قانون انتخاب همسایه ها به شکل مشابه به کار گرفته می شود. به طور مثال برای یک درخت سه مرتبه ای متناظر با تابع تبدیل g_1 ، هر نقطه ۷ همسایه از برگ های هم شاخه با خود، هفت همسایه سایر شاخه های مرتبه ۲ که در شاخه مرتبه ۱ با آن مشترک هستند و ۷ همسایه از سایر شاخه های مرتبه ۱ برای خود انتخاب می کند.

با چنین ساختار انتخاب همسایگی، اندازه جدول مسیر یابی $O(K)$ می باشد که در آن K تعداد سلسله مراتب درخت متناظر با شبکه است. به طور مثال برای برای تابع تبدیل g_1 ، تعداد عضوهای جدول مسیریابی $L + 7 \cdot K$ است که L تعداد مجموعه برگ های جدول مسیر یابی است. مجموعه برگ ها در فاز دوم مسیر یابی که جستجوی خطی درون ظرف هاست به کار می آیند. برای تابع g_1 تعداد سلسله مراتب در نگاشت معمولاً از مرتبه $O(\log_8^N)$ انتخاب می شود تا تعداد نقاط موجود در هر ظرف کم باشد. بنابراین اندازه جدول های مسیر یابی نیز از مرتبه $O(\log_8^N)$ است.

هنگام دریافت یک بسته، نقطه انتقال دهنده تعداد بیت های مشترک میان ظرف مقصد بسته و ظرف خودش مقایسه می کند. اگر مقصد بسته در همان ظرف نقطه انتقال دهنده قرار داشته باشد، بسته به نقطه ای در میان مجموعه برگ ها که به مقصد نزدیک تر از سایرین است ارسال می شود. در غیر اینصورت اگر تعداد بیت های مشترک میان مقصد بسته و نقطه مورد نظر K باشد، بسته به عضو A در سطر $k+1$ جدول مسیریابی منتقل می شود که A به ظرفی تعلق دارد که حداقل $k+1$ بیت مشترک با ظرف مقصد دارد. بنابراین بسته با انتقال به A یک گام به مقصد نزدیک تر می شود. به طور مثال در شکل ۱-۷ اگر نقطه متعلق به ظرف ۷۳ بسته ای داشته باشد که به یکی از ظرف های ۷۷-۷۰ تعلق داشته باشد، بسته به صورت مستقیم به آن ظرف ارسال می شود. بسته های دریافتی متعلق به ظرف های ۱۷-۱۰ به ظرف ۱۵ ارسال می شوند و بسته های متعلق به ظرف های ۷-۰ به ظرف شماره ۲ فرستاده می شوند.

همانطور که قبلاً نیز ذکر کردیم، بسته های منتقل شده میان نقاط که در نزدیکی مرز نواحی واقع شده اند با وجود نزدیک بودن در هنگام مسیریابی باید از گام های متعددی عبور کنند. به طور مثال در شکل ۱-۷ بسته های منتقل شده میان نقاط موجود در ظرف های ۷ و ۱۰ که موقعیتی نزدیک به هم دارند در هنگام مسیریابی باید از چندین گام عبور کنند. استفاده از چند جدول نگاشت مستقل به منظور حل این مشکل پیشنهاد می شود. در این روش نقاط به جای نگاشته شدن در یک مجموعه از ظرف ها، به دو مجموعه مستقل از ظرف ها نگاشته می شوند. اگر نزدیکی نقاط در یکی از نگاشت ها حفظ نشود، احتمال بالایی وجود دارد که در نگاشت دیگر این نزدیکی حفظ شود. بنابراین دو جدول مسیریابی در هر نقطه نگهداری می شود که هر یک متناظر با یکی از این نگاشت ها است. مسیریابی با توجه به هر دو این جدول ها انجام می شود.

در مسیریابی با توجه به دو جدول، الگوریتم تصمیم گیری ذکر شده در بالا برای دومین جدول مسیر یابی تکرار می شود و عضو B در سطر 'K دومین جدول مسیریابی پیدا می شود. پس از آن، از میان دو عضو A و B انتخاب شده در جدول مسیریابی هر کدام که در مرتبه پایین تری قرار داشته باشند، انتخاب می شوند. بسته هایی که باید به مجموعه برگ ها ارسال شوند، در میان اجتماع مجموعه برگی های جدول مسیر یابی به بهترین انتخاب ارسال می شوند.

شبه کد الگوریتم مسیر یابی پیشنهادی در شکل ۷-۲ نشان داده شده است. الگوریتم ذکر شده هنگامی که یک بسته به موقعیت مقصد P در ظرف های D_1 و D_2 نسبت به دو مجموعه نگاشت به نقطه M در ظرف های A_1 و A_2 می رسد. نمادهای زیر در شبه کد مورد استفاده قرار گرفته اند:

$R_i(j,k)$: عضو موجود در سطر j ام و ستون k ام در i امین جدول مسیر یابی

$L_i(j)$: j امین نقطه در مجموعه برگ های i امین جدول مسیر یابی

$shl(A,B)$: طول پیشوند مشترک میان بیت های A و B

$D(i)$: i امین بیت در رشته D

از آنجا که بیشتر پیام های به روز رسانی یک نقطه باید به مجموعه نقاط علاقمند به دریافت آن ارسال شود، مسیر یابی چند پخشی یکی از نیاز های مهم در هنگام ارسال پیام های به روز رسانی است. الگوریتم مسیریابی پیشنهادی می تواند با تغییرات کوچکی برای انجام چندپخشی با کارایی بالا به کار گرفته شود. این ارسال بدین صورت انجام می شود که هر بسته به جای انتقال به یک تک نقطه به یک یا چند ظرف چند پخشی می شود. زمانی که یک نقطه میانی بسته ای را که مقصد چند پخشی آن ظرفی به غیر از ظرف خودش است را دریافت می کند، آن را مانند یک بسته تک پخشی در مسیر رسیدن به ظرف مناسب هدایت می کند. در حالت هایی که مقصد چند پخشی یک بسته ظرفی باشد که نقطه میانی در

```

If ( (D1=A1) or (D2=A2) ) then
//route the packet to the closest leaves
Forward to Li(j) s.th. |P- Li(j)| is minimum , Di=Ai
Else
//Route the packet with respect to the grid that brings it closer to its destination
X<=shl(D1,A1)+1;
Y<=shl(D2,A2)+1;
If (R1(x, D1(x))≠null or R2(y, D2(y))≠null ) then
If (x=y) then
forward to Z s.th |P-Z| minimum ,  $Z \in \{ R1(x, D1(x)), R2(y, D2(y)) \}$ 
Else
If ( (x > y) and R1(x, D1(x)) ≠null )
forward to R1(x, D1(x))
Else
forward to R2(y, D2(y))
End if
End if
Else
//rare case where all routing table entries are null
Forward to Z s.th.  $\max(\text{shl}(D1,Z1), \text{shl}(D2,Z2))$  is maximum ,  $Z \in R1 \cup R2$ 
// where Z belongs to buckets Z1 and Z2 with respect to the first and second grid
End if
End if

```

شکل ۷-۲ شبه کد الگوریتم مسیریابی هندسی

آن واقع است، نقطه میانی بسته را به تمام عضوهای مجموعه برگ های آن هدایت می کند. برگ های دریافت کننده نیز بسته را به صورت مشابه به مجموعه برگ های خود ارسال می کنند. به دلیل جلوگیری از ایجاد حلقه در ارسال بسته ها، هر نقطه یک بسته چند بخشی را تنها یک بار ارسال می کند.

نکته مهم دیگر در اجرای موفق الگوریتم، فرآیند مسیریابی نگهداری و به روز رسانی جدول های مسیریابی است. به این منظور به صورت مداوم پیام هایی برای به روز رسانی همسایگان مبادله می شوند. زمانی که یک نقطه حرکت می کند، موقعیتش در مجموعه ظرف های نقاط تغییر می کند.

بنابراین نقطه جابجا شده باید اعتبار همسایه های قبلی اش را بررسی کند و ممکن است نیاز به تغییر برخی از آنها وجود داشته باشد. علاوه بر آن نقاط که نقطه مورد نظر را در جدول مسیریابی خود داشته اند نیز ممکن است نیاز به تغییر آن داشته باشند. در قسمت بعدی به جزئیات این به روز رسانی خواهیم پرداخت.

۵.۷. مدیریت تغییرات در شبکه

برای پشتیبانی تغییرات مختلف شبکه مانند حرکت، پیوستن نقاط جدید به شبکه و جدا شدن نقاط قبلی از شبکه، نیاز به به روز رسانی مداوم جدول های مسیریابی وجود دارد. به روز رسانی جدول های مسیریابی می تواند به شکل عمومی و یا انتخابی انجام شود. به روز رسانی مداوم و عمومی جدول های مسیریابی ساده ترین روش هماهنگ نمودن شبکه با تغییرات است. ولی این روش به روز رسانی هزینه زیادی را بر شبکه تحمیل می کند. بنابراین در این قسمت ما روش به روز رسانی انتخابی جدول های مسیریابی را شرح می دهیم.

در روش به روز رسانی انتخابی، هر نقطه سعی دارد تا هنگام حرکت خود در ضمن به روز رسانی جدول مسیریابی خودش پیام های به روز رسانی مناسب را به نقاط که نیاز به تغییر در جدول مسیریابی آنها وجود دارد، ارسال نماید.

همانطور که قبلا نیز اشاره کردیم، مرتبه های همسایه های مختلفی برای یک نقطه تعریف می شوند که در فرآیند به روز رسانی هر دسته از همسایه ها به شیوه خاص خود به روز رسانی می شود. همسایه های مرتبه اول یک نقطه، در اصطلاح به همسایه هایی گفته می شود که در تمام سلسله مراتب ۱ تا K در ظرفی مشابه با آن نقطه قرار دارد. به شکل مشابه همسایه های مرتبه ۲ یک نقطه، نقاط هستند که در سلسله مراتب ۱ تا $k-1$ در ظرفی مشابه با آن نقطه قرار دارند و تنها ظرف مرتبه K ام آنها متفاوت

است. همین تعریف به همسایه های مرتبه بالاتر نیز توسعه داده می شود. بر مبنای تعریف این همسایگی ها روش انتخابی به روز رسانی جدول های مسیر یابی نقاط در سه حال پیوستن یک نقطه جدید به شبکه، حرکت نقاط در شبکه و خروج یک نقطه از شبکه، شرح داده می شود.

۱.۵.۷. پیوستن نقطه جدید به شبکه

برای پیوستن هر نقطه به شبکه، نقطه جدید باید یکی از اعضای فعلی شبکه را بشناسد که این عضو به عنوان دروازه ورود نقطه تازه وارد به شبکه عمل می کند. نقطه تازه وارد ابتدا با تنها نقطه ای که در شبکه می شناسد ارتباط برقرار می کند و درخواست خود را برای پیوستن به شبکه همراه با مختصات محل وارد شدن به آن ارسال می کند. نقطه دروازه این درخواست را دریافت می کند و آن را روی شبکه به آدرس مقصدی که برابر با مختصات ورود نقطه تازه وارد است، ارسال می کند. این درخواست بر روی شبکه مسیر یابی می شود و به دست نقطه ای که نزدیکترین موقعیت را نسبت به نقطه تازه وارد دارد می رسد. این نقطه به احتمال زیاد در یک ظرف با نقطه تازه وارد قرار دارد و جدول های مسیر یابی آنها امکان داشتن عضو های مشترک زیادی را دارند. نزدیک ترین نظیر به نظیر تازه وارد پس از آن در پاسخ به این درخواست جدول مسیر یابی خود را برای نقطه تازه وارد ارسال می کند تا در ابتدا از آن استفاده کند و در مرور زمان آن را در صورت نیاز اصلاح نماید.

۲.۵.۷. حرکت نقاط

هنگامی که یک نقطه در محیط مجازی حرکت می کند، اطلاعات به روز رسانی مکان جدید باید به تمامی نقاط که این نقطه را به عنوان همسایه در جدول مسیر یابی خود داشته اند ارسال شود. با دریافت اطلاعات مربوط به مکان جدید نقطه جابجا شده همسایگان آن در مورد اینکه هنوز می توانند از

این نقطه در جدول مسیریابی خود استفاده کنند تصمیم می گیرند و در صورت لزوم به دنبال جایگزین برای آن می گردند. همسایگان نقطه جابجا شده بر مبنای مرتبه همسایگی میان آنها واکنش های متفاوتی را هنگام تغییر مکان آن انجام می دهند. اگر همسایه مورد نظر یک همسایه مرتبه اول نقطه جابجا شده باشد، تنها در صورتی نیاز به انجام تغییر وجود دارد که نقطه مورد نظر در اثر تغییر مکان از ظرف مرتبه اول خود خارج شده باشد. در غیر اینصورت همسایه مرتبه اول تنها در مجموعه برگ های خود به دنبال همسایگان مناسب برای نقطه جابجا شده می گردد و در صورت پیدا کردن کاندیداهای مناسب آنها را به نقطه جابجا شده اطلاع می دهد. در صورتی که نقطه از ظرف مرتبه اول خود خارج نشده باشد، تنها مجموعه برگ های آن ممکن است تغییر کنند و امکان استفاده از سایر همسایه های قدیمی وجود دارد.

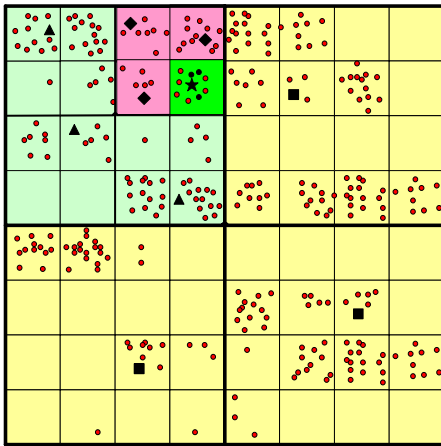
زمانی که یک نقطه ظرف مرتبه اول خود را تغییر می دهد، نیاز به جایگزینی مجموعه برگ ها و همسایه های مرتبه اول خود دارد. در چنین شرایطی همسایه های مرتبه دوم این نقطه در دو حالت قرار می گیرند: اگر نقطه جابجا شده به ظرف مرتبه اول متناظر با همسایه قبلی مرتبه دومش وارد شده باشد، این همسایه قبلی با توجه به اطلاعات جدول مسیر یابیش همسایه های مرتبه اول جدیدی را به آن نقطه پیشنهاد می دهد. در حالتی که نقطه جابجا شده به ظرف مرتبه اولی به غیر از ظرف همسایه مرتبه دومش داخل شده باشد، این همسایه مرتبه دوم نیاز به انجام کار خاصی ندارد. سناریوی مشابهی در مورد همسایگان مرتبه سوم یک نقطه در حالتی که نقطه ظرف مرتبه سوم خود را تغییر می دهد، اتفاق می افتد. نکته قابل توجه این است که تغییر در ظرف های مرتبه بالا که مستلزم تغییرات بیشتر در همسایگی ها است، به ندرت اتفاق می افتد.

شکل ۷-۳ فرآیند به روز رسانی همسایگی ها را در یک فضای نگاشت دو بعدی با سه مرحله از سلسله مراتب نشان می دهد. در اینجا سناریوی به روز رسانی را برای نقطه تصادفی در محیط مجازی

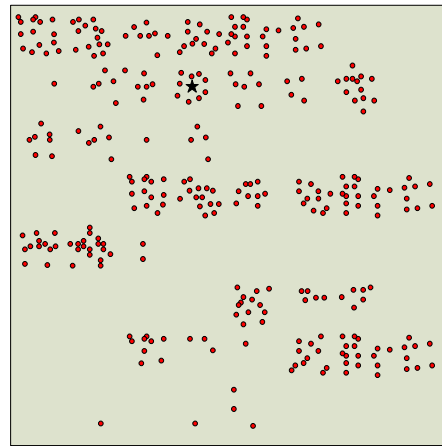
که با علامت ستاره در شکل ۳-۷-الف) نشان داده شده است، شرح می دهیم. در شکل ۳-۷-ب) موقعیت نقطه ستاره را نسبت به همسایگانش نشان می دهد. در این شکل همسایگان مرتبه اول با دایره ، همسایگان مرتبه دوم با لوزی ، همسایه های مرتبه سوم با مثلث و همسایه های مرتبه چهارم با مربع های های سیاهرنگ نشان داده شده اند. در شکل ۳-۷-ج) سناریوی حرکت نقطه ستاره در ظرف مرتبه اولش نشان داده شده است. نقطه ستاره در این شرایط با کمک همسایگان مرتبه اولش ، برگ های جدید را برای جدول مسیر یابی خود پیدا می کند. دایره های سفید رنگ در شکل نشان دهنده همسایه های قبلی نقطه ستاره هستند.

در شکل ۳-۷-د) سناریوی دیگری نشان داده شده است که در آن نقطه ستاره از ظرف مرتبه اول خود خارج می شود ولی هنوز داخل ظرف مرتبه دوم خود باقی می ماند. در چنین شرایطی نقطه ستاره باید همسایگان مرتبه اول خود را بر مبنای اطلاعات دریافت شده از همسایه قبل مرتبه دوم خود که اکنون با آن در یک ظرف قرار دارد، به روز رسانی می کند.

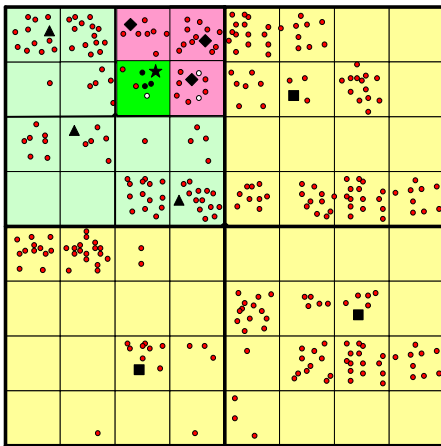
سناریوهای مشابهی در شکل ۳-۷-ه) و در شکل ۳-۷-و) اتفاق می افتد که در آنها نقطه ستاره از ظرف های مرتبه دوم و سوم خود خارج می شود. در همه شکل ها همسایه های سابق با رنگ سفید و همسایه های فعلی با رنگ سیاه نشان داده شده اند.



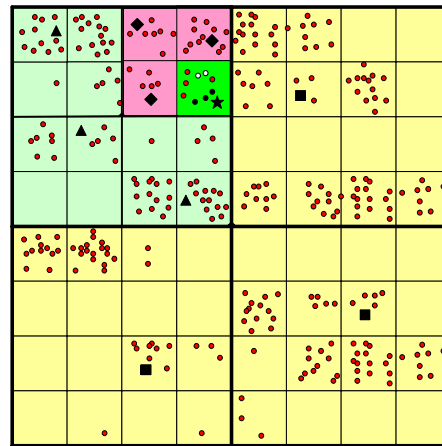
(ب) نقطه ستاره و همسایه های آن



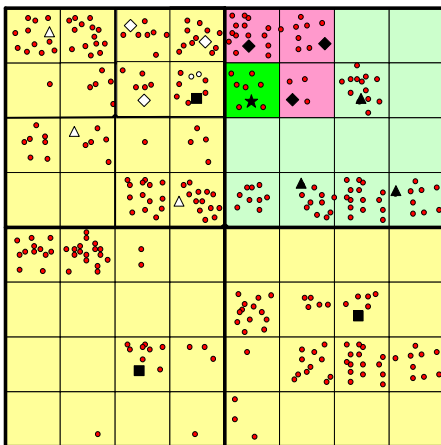
(الف) فضای دو بعدی بازی و اوتارهای آن



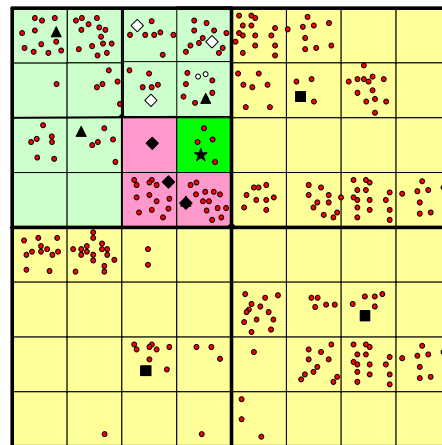
(د) ستاره ظرف مرتبه اول خود را تغییر می دهد



(ج) نقطه ستاره در ظرف خود حرکت می کند



(و) ستاره ظرف مرتبه سوم خود را تغییر می دهد



(ه) ستاره ظرف مرتبه دوم خود را تغییر می دهد

شکل ۳-۷ تغییر در همسایگی ها هنگام حرکت یک نقطه

۳.۵.۷. خارج شدن یک نقطه از شبکه

فرآیند خارج شدن یک نقطه از شبکه به سادگی صورت می پذیرد و نقطه خارج شونده بدون هیچ اطلاعی می تواند شبکه را ترک کند. سایر نقاط شبکه در هنگام تبادل مداوم پیام های به روز رسانی متوجه خروج نقطه مورد نظر می شوند و اگر این نقطه عضو جدول مسیر یابی آنها باشد، برای آن جایگزین مناسبی پیدا می کنند.

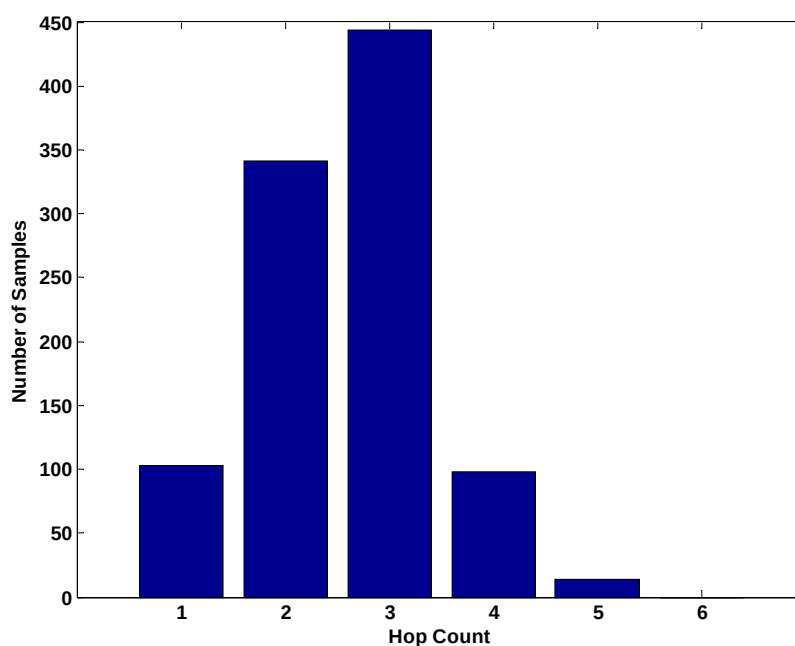
۶.۷. نتایج شبیه سازی

در این قسمت نتایج حاصل از شبیه سازی ساختار پیشنهادی را ارائه خواهیم کرد. توزیع کاربران در فضای محیط مجازی مشابه توزیع ذکر شده در فصل سوم است که تعداد کاربران آن ۱۰۰۰۰ در نظر گرفته شده است. انتخاب تعداد سلسله مراتب LSH بسته به چگالی کاربران در محیط مجازی دارد. انتخاب سلسله مراتب بیشتر سبب افزایش تعداد ظرف های نگاشت می شود و تعداد نقاط موجود در هر ظرف در نتیجه آن کاهش می یابد. کم بودن تعداد نقاط یک ظرف موجب تسریع در فاز دوم مسیر یابی که جستجوی خطی درون آن ظرف است، خواهد شد. اما از سوی دیگر افزایش تعداد ظرف های نگاشت باعث افزایش اندازه جدول های مسیر یابی و رشد متعاقب هزینه تبادل پیام های کنترل توپولوژی خواهد شد. در شبکه اول که بر مبنای تابع g_1 شکل می گیرد، یک تقریب کلی برای انتخاب سلسله مراتب $O(\log_8^N)$ است که در آن تعداد نقاط هر ظرف محدود خواهد بود، به صورت مشابه برای شبکه دوم تقریب تعداد سلسله مراتب $O(\log_{27}^N)$ خواهد بود. حدود این مقادیر در شبکه اول 5 و در شبکه دوم 3 است. برای سادگی مرتبه هر دو جدول مسیر یابی یکسان در نظر گرفته می شود. بنابراین در شبیه سازی های این قسمت تعداد سلسله مراتب برای هر دو شبکه، 5 در نظر گرفته شده است تا

برای هر دو شبکه مناسب باشد. در زیر بخش های آینده سناریو های شبیه سازی مختلفی را به منظور بررسی خاصیت های مختلف الگوریتم پیشنهادی ارایه خواهیم کرد.

۱.۶.۷. کارایی مسیریابی

شکل ۷-۴ نتایج شبیه سازی تعداد گام ها را در شرایط ذکر شده نشان می دهد. نتایج ثبت شده در بازه 1 ثانیه ای برای یک نقطه تصادفی می باشد. در این شکل متوسط تعداد گام ها در ارسال بسته ها ۲.۶ است و بیشترین تعداد گام ۵ است. در حالیکه بیشتر پیام ها در کمتر از ۴ گام به مقصد رسیده اند. همچنین در شکل دیده می شود که تعداد بسته هایی که در یک گام به مقصد رسیده اند کمتر از تعداد بسته هایی است که در دو گام به مقصد رسیده اند تعداد این بسته ها از تعداد بسته هایی که در سه گام به مقصد رسیده اند بیشتر است. و این تعداد برای پیام های ارسال شده که چهار گام یا بیشتر را در مسیر خود طی کرده اند به میزان قابل توجهی کاهش یافته است که دلیل آن را در فصول گذشته شرح دادیم.

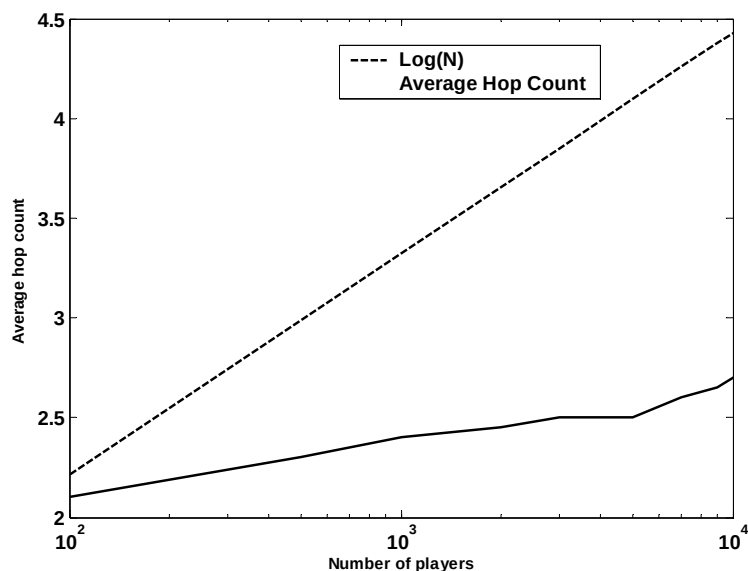


شکل ۷-۴ هیستوگرام گام های ارسال بسته ها

۲.۶.۷. قابلیت توسعه

در این قسمت ارتباط میان گام های مسیر یابی را با تعداد کاربران محیط مجازی بررسی خواهیم نمود. به این منظور تعداد کاربران محیط مجازی از ۱۰۰ تا ۱۰۰۰۰ تغییر داده شده است. در این شبیه سازی از یک شبکه استفاده شده است تعداد سلسله مراتب در جدول مسیر یابی هر مرحله برای هر دو شبکه شطرنجی برابر با $\lfloor \log_8^N \rfloor$ در نظر گرفته شده است. در هر مرحله از شبیه سازی مقدار متوسط تعداد گام های مسیر در شبکه محاسبه و ثبت شده است. شکل ۵-۷ متوسط تعداد گام ها را در هر مرحله در مقابل تعداد نقاط شبکه در آن مرحله نشان می دهد. همانطور که قبلا اشاره کردیم، اندازه جدول مسیر یابی در هر مرحله $O(\log_8^N)$ است و حداکثر تعداد گام ها در مسیر یک بسته $\log_8^N + L$ است که L تعداد گام های بسته در هنگام حرکت در ظرف مقصد برای رسیدن به نقطه مقصد است. بنابراین L بستگی چندانی به ابعاد شبکه ندارد و مقدار معمول آن صفر یا یک است در حالیکه $O(\log_8^N)$ به صورت مستقیم به ابعاد شبکه مرتبط است.

در شکل ۵-۷ دیده می شود که متوسط تعداد گام ها در شبکه شیب بسیار کمتری نسبت به $\log_8^N$ دارد. این ویژگی مطلوب به علت خواص حفظ نزدیکی در هنگام مسیر یابی بسته ها است که در آن ترافیک محلی می تواند بدون توجه به ابعاد شبکه در همسایگی نزدیک خود منتقل شود.



شکل ۵-۷ متوسط تعداد گام های مسیر بسته ها بر حسب تعداد نقاط شبکه

۳.۶.۷. حرکت نقاط

در آخرین سناریوی شبیه سازی عملکرد و کارایی الگوریتم به روز رسانی انتخاب همسایگی های شبکه مورد بررسی قرار می گیرد. شبیه سازی در یک محیط مجازی با ۱۰۰۰۰ نقطه انجام شده است که نقاط آن با سرعت تصادفی در بازه $[0.1 \ 0.2]$ units/sec و در جهت تصادفی در حال حرکت هستند. در این شرایط، الگوریتم به روز رسانی همسایه ها که در فصل قبل شرح داده شد، در شبکه اجرا می شود.

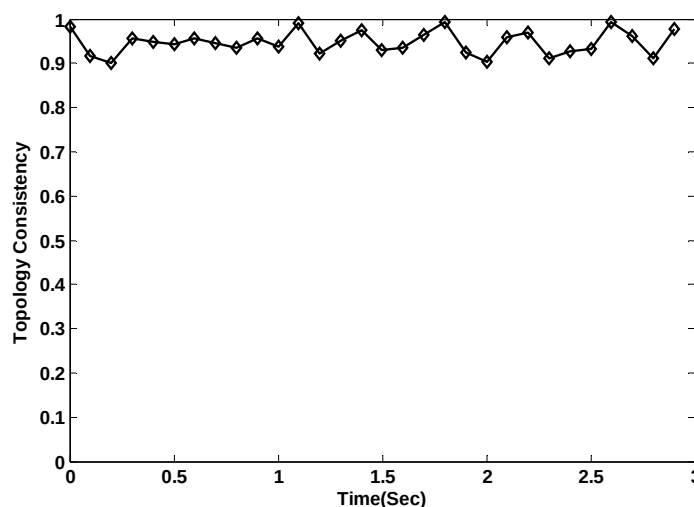
به منظور بررسی کارایی الگوریتم پیشنهادی معیاری را به عنوان سازگاری توپولوژی^۱ تعریف می کنیم. برای هر نقطه در شبکه و در هر لحظه از زمان، سازگاری توپولوژی برای آن نقطه به صورت نسبت همسایه های معتبر در جدول مسیر یابی آن به کل تعداد همسایه ها تعریف می شود. پهنای باند در دسترس نقاط برای انتقال پیام های به روز رسانی در شبکه به مقدار حداکثر 5kpbs محدود شده

^۱ topology consistency

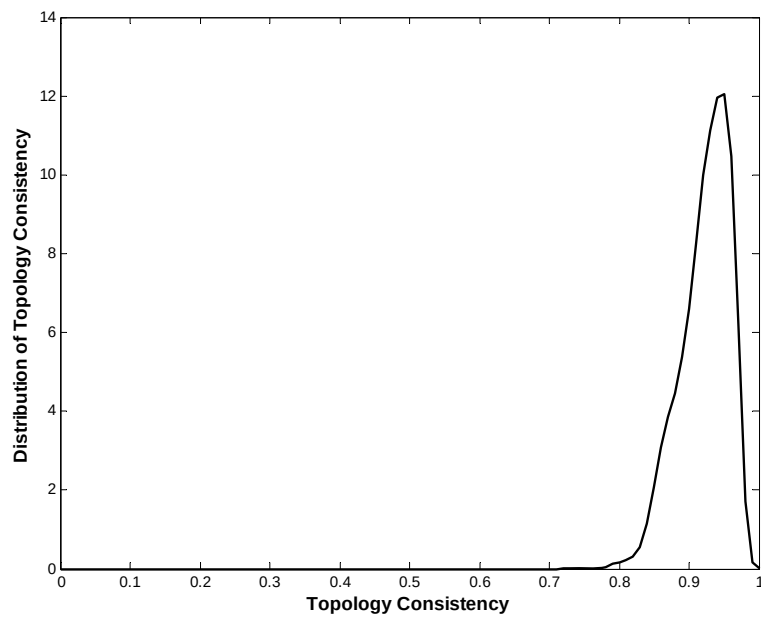
است.

شکل ۶-۷ مقدار سازگاری توپولوژی را در یک بازه زمانی برای یک نقطه تصادفی در شبکه نشان می دهد. همانطور که در شکل دیده می شود، سازگاری توپولوژی در متوسط 95% از مواقع حفظ می شود. در 5% موارد باقیمانده بسته ها به همسایه هایی منتقل می شوند که در رمویت قبلی خود قرار ندارند و بنابراین با گام های اضافه ای در مسیر حرکت خود روبرو می شوند ولی کماکان به نقطه مقصد خود هدایت خواهند شد.

به منظور نشان دادن یک شکل کلی تر از سازگاری توپولوژی، توزیع مقادیر سازگاری توپولوژی در کل شبکه در شکل ۷-۷ نشان داده شده است که در آن شرایط شبیه سازی مشابه قسمت قبل در نظر گرفته شده است. همانطور که در شکل دیده می شود، مقدار سازگاری در شبکه به صورت مطلق بیشتر از ۰.۸ است و مقدار میانگین آن در حدود ۰.۹۴ است. این مقادیر کارایی مناسب الگوریتم به روز رسانی همسایه ها را نشان می دهد.



شکل ۶-۷ سازگاری توپولوژی برای یک نقطه تصادفی



شکل ۷-۷ توزیع سازگاری توپولوژی در تمام شبکه

فصل هشتم
نتیجه گیری و پیشنهادات

با رشد تعداد کاربران محیط های مجازی استفاده از ساختارهای نظیر به نظیر به عنوان راه حلی برای گریز از نیاز به سرورهای بسیار قدرتمند، پیشنهاد شده است. چالش اصلی در ساختارهای گسترده حل غیر متمرکز مسایل مدیریتی است. علاوه بر ساختارهای نظیر به نظیر عموماً از مسیر یابی در لایه کاربرد استفاده می کنند که منجر به افزایش تأخیر در ارسال بسته ها خواهد شد، افزایش این تأخیر می تواند اثرات بسیار نامطلوبی در تعامل کاربران ایجاد نماید. بنابراین استفاده از ساختارهای نظیر به نظیر نیازمند راه کارهایی برای جبران سازی این تأخیر است. در این رساله در فصول مختلف ساختارهای گسترده متعددی را برای محیط های مجازی گسترده پیشنهاد دادیم که هدف در طراحی تمامی آنها فراهم سازی امکان قابلیت توسعه و کمینه سازی تأخیر انتقال پیام های به روز رسانی است. وجه مشترک میان تمام این ساختارها تشکیل یک شبکه overlay بر روی اینترنت با توجه به الگوی انتقال ترافیک است. در این رساله ساختارهای زیر را برای محیط های مجازی ارائه نمودیم :

- نخستین ساختار پیشنهادی از شبکه های DHT که در حوزه شبکه های نظیر به نظیر تقسیم فایل

بسیار مشهور هستند، الهام گرفته است. در عین حال این ساختار در جنبه هایی اساسی با DHT متفاوت است و برای استفاده در کاربرد محیط های مجازی تطبیق یافته است. نخستین تغییر در ساختار اصلی DHT شیوه تخصیص شناسه به نقاط است که در ساختار پیشنهادی به جای استفاده از توابع متداول hashing، مختصات سه بعدی نقاط با کمک منحنی هیلبرت به یک شناسه یک بعدی نگاشته می شود. بدین ترتیب فاصله نقاط در محیط مجازی در ساخت شبکه overlay منعکس می شود. الگوریتم مسیر یابی پیشنهادی از لحاظ ساختار اصلی مشابه الگوریتم مسیر یابی Pastry است. ولی به منظور کمینه کردن تأخیر مسیر ها سیاست های جدیدی در هنگام انتخاب همسایه ها و انتخاب مسیر های انتقال بسته ها به کار گرفته می شود. به منظور جستجوی همسایه های نزدیک و پیدا کردن کوتاه ترین مسیر ها برای پیام های به روز رسانی از مورچه هایی استفاده می شود که بر روی پیام های به روز رسانی سوار می شوند. این مورچه ها در تمامی شبکه منتشر شده و جدول مسیر یابی نقاط را به روز رسانی می کنند.

- ساختار پیشنهادی دوم بر مبنای منحنی هیلبرت می باشد. در این روش تخصیص شناسه به نقاط و مسیر یابی بر مبنای منحنی هیلبرت انجام می شود که انگیزه انتخاب منحنی هیلبرت خواص بسیار مناسب آن در حفظ نسبت فاصله نقاط هنگام انجام نگاشت است. در این ساختار مختصات سه بعدی نقاط با کمک تبدیل هیلبرت به یک بعد نگاشته می شود و به منظور مسیر یابی بسته ها میان نقاط از مدل درختی منحنی هیلبرت استفاده می شود. یک منحنی هیلبرت سه بعدی از مرتبه k می تواند به شکل درختی با k مرتبه نمایش داده شود. هر نقطه محیط مجازی یک برگ از این درخت است و از زیر شاخه های مناسب درخت برای خود همسایگانی را انتخاب و نگهداری کند. بدین ترتیب، هر نقطه امکان انتقال هر بسته ای را به شکلی که بسته

مورد نظر یک شاخه به مقصدش نزدیک تر شود، خواهد داشت. ساختار هیلبرت با توجه به الگوی محلی تبادل پیام ها در محیط مجازی می تواند مقادیر تأخیر ارسال را به میزان قابل توجهی کاهش دهد.

- ساختار سوم پیشنهادی بر مبنای نگاشت شناسه نقاط با منحنی Z و مسیر یابی آنها بر روی درخت KD تعریف شده است. استفاده از نگاشت Z مشابه با نگاشت هیلبرت به دلیل خواص مناسب آن در حفظ نسبت فاصله نقاط هنگام نگاشت است. ایده استفاده از درخت KD از شهرت آن در حوزه پایگاه های داده الهام می گیرد. نقاط محیط مجازی را می توان به صورت رکوردهای یک پایگاه داده گسترده تصور نمود که مسیریابی پیام های به روز رسانی به آنها مشابه ارسال درخواست ها بر روی پایگاه داده است. به همین دلیل استفاده از درخت KD به عنوان یک ابزار مناسب برای جستجوی سریع اطلاعات در یک پایگاه داده چند بعدی، انتخاب مناسبی به نظر می رسد. همچنین درخت KD برای اجرای درخواست های ناحیه ای نیز بسیار کارآمد است که این نوع درخواست ها به دلیل نیاز به ارسال پیام های به روز رسانی به نواحی پیوسته در فضا، بسیار مورد استفاده هستند.

- آخرین ساختار پیشنهادی برای محیط های مجازی بر مبنای مسیر یابی هندسی تعریف شده است. از آنجا که پیام های به روز رسانی باید به موقعیت نقاط در محیط مجازی ارسال شوند، استفاده از نوعی از سرویس مسیریابی که بسته ها را به جای یک آدرس IP به یک مختصات مشخص منتقل کند در این کاربرد بسیار مطلوب به نظر می رسد و نیاز به استفاده از سرویس مکان یابی را برطرف می کند. مسیر یابی هندسی که در ابتدا در حوزه شبکه های بیسیم تعریف شده است چنین قابلیتی را فراهم می آورد. ولی مسیر یابی هندسی در شکل اصلی تعریف شده

برای کاربرد محیط های مجازی مناسب نمی باشد. بنابراین تغییرات مختلفی در ساختار پیشنهادی برای تطبیق آن پیشنهاد شده است که از جمله آنها ساختار سلسله مراتبی مسیریابی و مسیر یابی بر روی چندین شبکه به شکل همزمان می باشد.

ساختارهای پیشنهادی هر یک مزایا و معایب خاص خود را به همراه دارند. کلیات این مقایسه در جدول ۸-۱ آمده است. نخستین معیاری که در طراحی الگوریتم ها مورد توجه بوده است قابلیت توسعه پذیری می باشد. همانطور که قبلاً هم اشاره نمودیم، نخستین انگیزه در استفاده از ساختارهای نظیر به نظیر قابلیت توسعه پذیری این ساختارها می باشد که در اینجا هر چهار ساختار پیشنهادی چنین قابلیت را دارا می باشند. در مورد اندازه جدول مسیر یابی نیز نکته قابل توجه آن است که تمام ساختارها جداول مسیر یابی از مرتبه $O(\log(N))$ دارند که معمول ساختارهای DHT نیز می باشد. ولی همانطور که در شبیه سازی ها نشان داده شده است، مرتبه تعداد گام های مسیر یابی از $O(\log(N))$ که معمول شبکه های DHT است بسیار کمتر می باشد. بنابراین ساختارهای پیشنهادی با ثابت نگاه داشتن مرتبه جدول مسیر یابی تعداد گام های مسیر یابی را کم کرده اند.

مقایسه دیگری که میان این ساختارها می تواند صورت گیرد پیچیدگی محاسباتی در نگاشت شناسه ها به نقاط است. این پیچیدگی در نگاشت شناسه هیلبرت به یک منحنی هیلبرت k بعدی از مرتبه n معادل $O(nk)$ [84] و برای این محاسبه به نگهداری یک جدول حالت از مرتبه 2^{n-1} و تعداد n جستجو در آن است. در منحنی Z از مرتبه k بعدی از مرتبه n نگاشت تنها به $O(nk)$ محاسبه نیاز دارد. بنابراین استفاده منحنی Z به مراتب ساده تر از منحنی هیلبرت است. در ساختارهای پیشنهادی $k=3$ برابر با ابعاد محیط مجازی است و n در حدود $\log(N)$ در نظر گرفته می شود که N تعداد کاربران محیط مجازی است. در مسیر یابی هندسی پیچیدگی تشخیص شناسه وجود ندارد و از مختصات نقاط به عنوان

شناسه استفاده می شود.

بهینه سازی تأخیر نیز یکی دیگر از اهداف مهم در طراحی این ساختارها بوده است. در این زمینه با توجه به متوسط تأخیر در سناریو های شبیه سازی قسمت قبل، می توان این چهار روش را بر مبنای می نیمم سازی تعداد گام ها در مسیر یابی، به شکل نشان داده شده در جدول ۸-۱ رده بندی نمود که در این رده بندی کارایی ساختار هندسی از همه بهتر است. نکته قابل توجه در این مقایسه آن است که روشهای جستجوی مسیرها و همسایگان نزدیک در هر چهار ساختار قابل اعمال است و ما تنها این روش ها را برای نمونه در ساختار اول اعمال نمودیم. به منظور مقایسه مقادیر مطلق تأخیر میان این چهار ساختار، این روش باید در سایر ساختارها نیز اعمال شود.

جدول ۸-۱ مقایسه کلی و رده بندی ساختارهای پیشنهادی

نوع ساختار	توسعه پذیری	اندازه جدول مسیریابی	هزینه پیاده سازی	کمینه سازی تأخیر
DHT	توسعه پذیر	$O(\log(N))$	محاسبه: $O(3\log(N))$ حافظه: $\log(N)2^{\log(N)-1}$ جستجو: $\log(N)$	۴
هیلبرت	توسعه پذیر	$O(\log(N))$	محاسبه: $O(3\log(N))$ حافظه: $\log(N)2^{\log(N)-1}$ جستجو: $\log(N)$	۳
درخت KD	توسعه پذیر	$O(\log(N))$	محاسبه: $O(3\log(N))$	۲
هندسی	توسعه پذیر	$O(\log(N))$	۰	۱

با توجه به تحقیقات انجام شده در رساله، زمینه های تحقیقی زیر به عنوان کارهای آینده پیشنهاد می شوند :

- نکته قابل توجه در مورد ساختارهای پیشنهادی آن است که تمامی آنها با تغییراتی قابلیت به کار

گیری در شبکه های گسترده با ساختار غیرمتمرکز که الگوی ارسال ترافیک در آنها بیشتر به شکل محلی است، را دارند. از این رو حوزه پوشش این رساله می تواند در کارهای آینده به سایر کاربردها مانند شبکه های سنسور بی سیم تعمیم داده شود. یکی از زمینه های کاری آینده در این رساله به کارگیری این الگوریتم ها در کاربردهای مختلف مشابه است.

- نخستین گام در طراحی یک ساختار مناسب برای یک محیط مجازی طراحی یک الگوریتم مدیریت ناحیه مورد علاقه به شکل گسترده است. در این رساله ما یک الگوریتم مدیریت ناحیه مورد علاقه ساده را پیشنهاد نمودیم که الگوی ترافیک مفروض در سایر قسمت ها بر مبنای این الگوریتم بوده است. با وجود آنکه تمرکز این رساله بر روی طراحی الگوریتم های مدیریت ناحیه مورد علاقه نبوده است، این حوزه نیز یکی از زمینه های پر اهمیت در کاربر محیط های مجازی است.

- در فصل پنجم این رساله، ساختاری را بر مبنای منحنی هیلبرت پیشنهاد دادیم. منحنی هیلبرت فضا را به شکل یکنواخت جاروب می کند و چگالی نقاط آن در تمامی فضا یکسان است. این در حالی است که در محیط مجازی معمولاً چگالی نقاط در نزدیکی صفحه $z=0$ بیشتر است. بنابراین در این شرایط استفاده از فرم تعمیم یافته ای از منحنی هیلبرت که چگالی نقاط آن یکنواخت نباشد می تواند مورد بررسی قرار بگیرد.

- در روش های مبتنی بر درخت KD و ساختار هندسی، کمینه سازی تعداد گام های مسیر مورد توجه بوده است. در کنار این ساختارها الگوریتم هایی نیز می توانند برای انتخاب گام های کوتاه به منظور کمینه سازی تأخیر مسیرها مورد استفاده قرار بگیرند.

- در الگوریتم های پیشنهادی در این رساله، تأخیر به عنوان معیار کیفیت سرویس در ارسال بسته

ها فرض شده است و همانطور که قبلاً اشاره نمودیم فرض بر آن بوده است که لغزش با بافر کردن ترافیک و مصالحه تأخیر قابل جبران سازی است. بنابراین طراحی الگوریتم هایی که سعی در کمینه سازی همزمان تأخیر و لغزش داشته باشند در کارهای آینده قرار می گیرد. نکته قابل توجه آن است که لغزش اثر نامطلوب تری نسبت به تأخیر بر تعامل کاربران در محیط مجازی دارد. ولی در عین حال به علت عدم وجود خاصیت جمع پذیری برای لغزش در طول یک مسیر، کمینه سازی آن بسیار پیچیده تراز تأخیر است.

- ساختارهای ارایه شده در واقع نمونه هایی از نسل دوم شبکه های نظیر به نظیر هستند که علاوه بر مدیریت منابع گسترده، باید کیفیت سرویس را هم در هنگام انتقال پیام ها فراهم آورند. تحقیقات آینده می تواند در جهت تکامل این سیستم ها حرکت کنند.

در نهایت اینکه تلاش ما بر آن بوده است که در حد بضاعت زمانی محدود این تحقیق به توسعه ساختارهای مناسب برای محیط های مجازی گسترده بپردازیم. امید بر آن است که این تحقیق بتواند پیش زمینه مناسبی را برای تحقیقات آینده فراهم آورد و به نوبه خود به پیشرفت در این حوزه کمک کند.

مراجع

- [1] D.C. Miller and J.A. Thorpe, "SIMNET: The Advent of Simulator Networking," Proc. of the IEEE special Issue on Distributed Interactive Simulation, vol 83, no. 8, 1995, pp. 1114-1123.
- [2] IEEE Computer Society, IEEE Std 1278.1a-1998: IEEE standard for distributed interactive simulation - application protocols. 1998: New York.
- [3] M.R. Macedonia, M. J. Zyda , D. R. Pratt , P. T. Barham and S. Zeswitz, "NPSNET: A Network Software Architecture for Large Scale Virtual Environments," Presence, vol. 3, no. 4, 1994.
- [4] Distributed Systems Group, The PARADISE Project. 2004, Stanford University: Stanford, CA. <http://www.dsg.stanford.edu/paradise.html>
- [5] C. Greenhalgh and S. Benford, "MASSIVE: a collaborative virtual environment for teleconferencing," ACM Trans. on Computer-Human Interaction (TOCHI), 1995, pp. 239-261.
- [6] E. Lety, L. Gautier, and C. Diot. Mimaze, "A 3D multi-player game on the internet," Proc. of the 4th International Conf on Virtual System and Multimedia, vol. 1, 1998, pp. 84-89.

- [7] C. Carlsson and O. Hagsand, "DIVE - A Multi-user virtual reality system," Proc. of the Virtual Reality Annual International Symposium, 1993.
- [8] E. Frecon and M. Stenius, "DIVE – a scalable network architecture for distributed virtual environments," Distributed Systems Engineering Journal (Special issue on Distributed Virtual Environments), vol. 5, 1998.
- [9] T.A. Funkhouser, "RING: A Client-Server System for Multi-User Virtual Environments," ACM SIGGRAPH Special Issue on Symposium on Interactive 3D Graphics, 1995.
- [10] G. Singh, G., L. Serra, W. Png, A. Wong and H. Ng, "BrickNet: Sharing Object Behaviors on the Net," Proc. of the Virtual Reality Annual International Symposium (VRAIS'95), 1995, pp. 19-25.
- [11] Market Forecast : DFC Intelligence : www.dfciint.com
- [12] H. Balakrishnan H., M.F. Kaashoek, D. Karger, R. Morris and I. Stoica, "Looking up data in P2P systems," Communications of the ACM, vol. 46 no.2, 2003.
- [13] B. Moon, H.V. Jagadish, C. Faloutsos, J.H. Saltz, "Analysis of the Clustering Properties of the Hilbert Space-Filling Curve," IEEE Trans. Knowledge and Data Engineering, 2001, pp. 124—141.
- [14] J.L. Bentley, "Multidimensional binary search trees in database applications," IEEE Trans. on Software Engineering, vol. 5, no. 4, 1979.
- [15] I. Clarke, O. Sandberg, B. Wiley and T.W. Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System," Lecture Notes in Computer Science, 2000, pp. 44-46.
- [16] I.J. Taylor, and A.B. Harrison, "Gnutella," Springer Computer Communications and Networks, From P2P and Grids to Services on the Web, 2009, pp. 181-196.
- [17] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, "A scalable content-addressable network," Proc. of ACM SIGCOMM, 2001, pp. 161–172.

- [18] P. Druschel and A. Rowstron, “Pastry: Scalable, distributed object location and routing for large-scale peer-to peer systems,” Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), 2001.
- [19] P. Maymounkov and D. Mazières, “Kademlia: A Peer-to-Peer Information System Based on the XOR Metric,” Revised Papers from the First International Workshop on Peer-to-Peer Systems, 2002, pp.53-65.
- [20] I. Stoica, R. Morris, D. Karger, MF. Kaashoek and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” Proc. of the ACM SIGCOMM, 2001, pp. 149 – 160.
- [21] B. Zhao, J. Kubiatowicz and A. Joseph, “Tapestry: An infrastructure for fault-tolerant wide-area location and routing,” Tech. Rep. UCB/CSD-01-1141, University of California at Berkeley 2001.
- [22] J. Keller and G. Simon, “Solipsis: A massively multi-participant virtual world,” Proc Int. Conf. Parallel and Distributed Techniques and Applications, 2003, pp. 262–268.
- [23] H. Backhaus, S. Krause, “ QuON – a Quad-Tree Based Overlay Protocol for Distributed Virtual Worlds”, Proc. Massively Multiuser Virtual Environments (MMVE '09) ,2009.
- [24] E. Cronin, B. Filstrup, A. R. Kurc, and S. Jamin, “An efficient synchronization mechanism for mirrored game architectures,” Proc. NetGames '02, 2002, pp. 67–73.
- [25] S. Yamamoto, Y. Murata, K. Yasumoto, and M. Ito, “A distributed event delivery method with load balancing for mmorpg,” Proc. 4th ACM SIGCOMM workshop on Network and system support for games(NETGAMES), 2005, pp. 1-8.
- [26] L. Fan, H. Taylor, and P.W. Trinder, “Design Issues for Peer-to-Peer Massively Multiplayer Online Games,” Proc. Massively Multiuser Virtual Environments (MMVE '09), 2009.

- [27] S. Krause, “A Case for Mutual Notification: A Survey of P2P Protocols for Massively Multiplayer Online Games,” in *Proceedings of NetGames 2008 Network and Systems Support for Games*, 2008, pp. 28–33.
- [28] S. Benford S., L. Fahlen, “A spatial model of interaction in large-scale virtual environments,” *Proc. of the 3rd European Conference on CSCW*, 1993, pp. 109-124.
- [29] N. Matsumoto, Y. Kawahara, H. Morikawa, and T. Aoyama, “A Scalable and Low Delay Communication Scheme for Networked Virtual Environments,” *Proc. of IEEE GLOBECOM*, 2005, pp. 529-535.
- [30] A. Harwood, S. Karunasekera, S. Nutanong, E. Tanin, M. Truong, “Complex applications over peer-to-peer networks,” *Proc. of ACM Middleware*, 2004, pp. 327.
- [31] S. Douglas S, E. Tanin, A. Harwood, and S. Karunasekera, “Enabling Massively Multi-Player Online Gaming Applications on a P2P Architecture,” *Proc. of IEEE Intl. Conf. on Information and Automation*, 2005, pp. 7-12.
- [32] G. Morgan, F. Lu, and K. Storey, “Interest Management Middleware for Networked Games,” *Proc. ACM i3D*, 2005.
- [33] E. Buyukkaya and M. Abdallah, “Data Management in Voronoi-Based P2P Gaming,” *Proc. IEEE CCNC*, 2008, pp. 1050–1053.
- [34] S.-Y. Hu, S.-C. Chang, and J.-R. Jiang, “Voronoi State Management for Peer-to-Peer Massively Multiplayer Online Games,” *Proc. of IEEE CCNC. ,* 2008, pp. 1134–1138.
- [35] S.-Y. Hu, J.-F. Chen, and T.-H. Chen, “VON: A Scalable Peer-to-Peer Network for Virtual Environments,” *IEEE Network*, vol. 20, no. 4, 2006, pp.22–31.
- [36] S.-Y. Hu and G.-M. Liao, “Scalable Peer-to-Peer Networked Virtual Environment,” *Proc. of ACM NetGames*, 2004, pp. 129–133.
- [37] S. Fiedler, M. Wallner, and M. Weber, “A Communication Architecture for Massive Multiplayer Games,” *Proc. of ACM NetGames*, 2002, pp. 14–22.

- [38] S.Y. Hu, "Spatial Publish Subscribe," Proc. IEEE Virtual Reality (IEEE VR) workshop Massively Multiuser Virtual Environment (MMVE'09), 2009
- [39] T. Hampel, T. Bopp, and R. Hinn, "A Peer-to-Peer Architecture for Massive Multiplayer Online Games," Proc. of 5th ACM SIGCOMM workshop on Network and system support for games(NETGAMES), 2006, pp. 48.
- [40] M. Castro, P. Druschel, A.M. Kermarrec, and A.I.T Rowstron," Scribe: a large-scale and decentralized application-level multicast infrastructure," IEEE Journal on Selected Areas in Communications, vol. 20, no. 8, 2002, pp. 1489 – 1499.
- [41] H. Lu, B. Knutsson, W. Xu, and B. Hopkins, "Peer-to-Peer Support for Massively Multiplayer Games," Proc. of IEEE INFOCOM, 2004, pp. 7-11.
- [42] B. Knutsson, H. Lu, W. Xu, and B. Hopkins, "peer-to-peer Support for Massively Multiplayer Games," Proc. of IEEE INFOCOM, 2004, pp. 96-107.
- [43] A. Bharambe, J. Pang, and S. Seshan, "Colyseus: A Distributed Architecture for Multiplayer Games," Proc. ACM/USENIX NSDI, 2006, pp 3-12.
- [44] L. Fan L, H. Taylor, and P. Trinder, "Mediator: A Design Framework for P2P MMOGs," Proc. 6th ACM SIGCOMM workshop on Network and system support for games (NETGAMES), 2007, pp.43-48.
- [45] B. B. Yang and H. Garcia-Molina, "Designing a Super-Peer Network," Proc. IEEE ICDE, 2003, pp. 49–60.
- [46] M. Assiotis, and V. Tzanov, "A Distributed Architecture for MMORPG," Proc. of ACM NetGames, 2006.
- [47] M. R. Crippa, F. R. Cecin, C. F. R. Geyer, "Peer-to-peer support for instance-based massively multiplayer games", Proc. Brazilian Symposium on Computer Games and Digital Entertainment (SBGAMES'07)},2007.
- [48] L. Chan, J.Yong, J. Bai, B. Leong, R. Tan, "Hydra: A Massively-Multiplayer Peer-to-Peer Architecture for the Game Developer", Proc. ACM NetGames, 2007, pp. 14–22.

- [49] A. Yu, and ST. Vuong, "MOPAR: a Mobile Peer-to-Peer Overlay Architecture for Interest Management of Massively Multiplayer Online Games," Proc. NOSSDAV, 2005, pp. 99–104.
- [50] A. E. Rhalibi, and M.Merabti, "Interest Management and Scalability Issues in P2P MMOG," Proc. IEEE CCNC, 2006, pp. 1188-1192.
- [51] T. Limura, H. Hazeyama, and Y. Kadobayashi, "Zoned Federation of Game Servers: a Peer-to-Peer Approach to Scalable Multi-player Online Games," Proc. SIGCOMM, 2004, pp. 116-120.
- [52] C. G. Dickey, D. Zappala, and V. Lo, "A Fully Distributed Architecture for Massively Multiplayer Online Games," Proc. ACM NetGames, 2004, pp. 171.
- [53] A. Bharambe, M. Agrawal, and S. Seshan, "Mercury: Supporting scalable multi-attribute range queries," Proc. ACM SIGCOMM, 2004.
- [54] R. Simon, W. Klaus, F. Marc, N. Heiko, P. Leo, and C. Georg, "Peerto-Peer-Based Infrastructure Support for Massively Multiplayer Online Games," Proc. IEEE CCNC, 2007, pp. 763–767.
- [55] P. Morillo, W. Moncho, J. M. Ordua, and J. Duato, "Providing full awareness to distributed virtual environments based on peer-to-peer architectures," Springer Lecture Notes in Computer Science, Advances in Computer Graphics, vol. 4035, 2006, pp. 336–347.
- [56] J.-S. Boulanger, J. Kienzle, and C. Verbrugge, "Comparing Interest Management Algorithms for Massively Multiplayer Games," Proc. ACM NetGames, 2006.
- [57] N. Beatrice, S. Antonio, L. Rynson, and L. Frederick, "A multiserver architecture for distributed virtual walkthrough," Proc. ACM VRST'02, 2002, pp. 163-170.
- [58] J. Chim, R.W.H. Lau, H.V. Leong, and A. Si, "CyberWalk: A Web-based Distributed Virtual Walkthrough Environment," IEEE Trans. On Multimedia.

- [59] S. Rueda, P. Morillo, J.M. Orduna, and J. Duato, "On the Characterization of Peer-To-Peer Distributed Virtual Environments," Conf. Virtual Reality (VR apos'07) , 2007, pp.107 – 114.
- [60] M. Costa, M. Castro, A. Rowstron, and P. Key, "PIC:Practical Internet coordinates for distance estimation," Proc. ICDCS, 2004, pp. 178-187.
- [61] H. Lim, JC. Hou, and CH. Choi, "Constructing Internet coordinate system based on delay measurement," IEEE/ACM Trans. On Networking, vol. 13, no.3, pp. 513 – 525.
- [62] TSE. NG and H. Zhang, "Predicting Internet network distance with coordinates-based approaches," Proc. IEEE INFOCOM, vol. 1, 2002, pp. 170- 179.
- [63] R. Cox, F. Dabek, F. Kaashoek, J. Li and R. Morris, "Practical, distributed network coordinates," ACM SIGCOMM Computer Communication Review vol. 34, 2004, pp. 113-118.
- [64] B. Hoppe,O. Tardos ,” The Quickest Transshipment Problem,” Mathematics of Operations Research, vol. 25, no. 1, 2000, pp. 36-62.
- [65] RT. Rockafellar, Network flows and monotropic optimization. New York :Wiley; 1984.
- [66] RK. Ahuja, TL. Magnanti, and JB. Orlin, Network Flows: Theory, Algorithms, and Applications. New Jersey : Prentice Hall; 1993.
- [67] <http://www.ilog.com/products/cplex/>
- [68] F. Dabek, MF. Kaashoek, J. Li, R. Morris, J. Robertson, and E. Sit, "Designing a DHT for low latency and high throughput," Proc. the 1st NSDI, 2004, pp. 85-98.
- [69] M. Castro, P. Druschel, YC. Hu, and A. Rowstron, "Exploiting network proximity in distributed hash tables," Proc. FuDiCo, 2002, pp. 52–55.
- [70] M. Castro, P. Druschel, YC. Hu, and A. Rowstron, "Topology aware routing in structured peer-to-peer overlay networks," Tech. Rep. MSR-TR-2002-82, Microsoft Research, 2002.

- [71] M. Castro, P. Druschel, YC. Hu, A. Rowstron, "Proximity neighbor selection in tree-based structured peer-to-peer overlays," Technical report MSR-TR-2003-52, Microsoft Research, 2003.
- [72] J. Song, S. Park, J. Yang, "An Adaptive Proximity Route Selection Scheme in DHT-Based Peer to Peer Systems," LNCS, vol. 3320, 2005, pp. 778-781.
- [73] H. Zhang, A. Goel, and R. Govindan, "Improving lookup latency in distributed hash table systems using random sampling," IEEE/ACM Trans on Networking, vol. 13, no. 5, 2005, pp. 1121-1134.
- [74] P. Karwaczynski and J. Mocnik, "Self-optimization of a dht based discovery service," Proc. ICCGI, 2007, pp. 39-39.
- [75] GM. Morton, "A computer oriented geodetic data base and a new technique in file sequencing," TECH Report, IBM Ltd., Ottawa, Canada, 1966.
- [76] F. Gray F, "Pulse Code Communication," U.S. Patent 2,632,058, 1953.
- [77] G. Breinholt, C. Schierz, "generating Hilbert's space-filling curve by recursion," ACM Trans. on Mathematical Software (TOMS), vol. 24, no. 2, 1998, pp. 184-189.
- [78] HV. Jagadish, "Linear Clustering of Objects with Multiple Attributes," Proc. ACM SIGMOD, 1990, pp. 332-342.
- [79] B. Moon B, HV. Jagadish, C. Faloutsos, and J. Salz, "Analysis of the Clustering Properties of Hilbert Space-filling Curve," IEEE Trans. on Knowledge and Data Engineering (IEEE-TKDE), vol 13, no. 1, 2001, pp. 124-141.
- [80] M. Dorigo, V. Maniezzo, A. Colorni, "Ant System: Optimization by a colony of cooperating agents," IEEE Trans. on Systems, Man, and Cybernetics, 1996, pp. 29-41.
- [81] R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rosenkrantz, "Ants for load balancing in telecommunication networks," Technical Report 96-35, HP Labs, Bristol, 1996.

- [82] K.S. Park, R.V. Kenyon, "Effects of Network Characteristics on Human Performance in a Collaborative Virtual Environment," *proc. VR*, 1999, pp. 104.
- [83] J.K. Lawder, "The Application of Space-Filling Curves to the Storage and Retrieval of Multi-dimensional Data," PhD Thesis, Birkbeck College, University of London, 1999.
- [84] J.K. Lawder, "Using State Diagram for curve mapping," *International Journal of Computer Mathematics*, vol. 78, no. 3, 2001, pp. 327-342.
- [85] J.K. Lawder, P.J.H. King, "Using space-filling curves for multi-dimensional indexing," *Lecture Notes in Computer Science*, Vol. 1832, 2000, pp. 20 – 35.
- [86] H. Tropf, H. Herzog, "Multidimensional range search in dynamically balanced trees," *Applied Informatics*, Vieweg Verlag, Wiesbaden, Germany, 1981, pp. 71-77.
- [87] G. M. Morton, "A computer oriented geodetic data base; and a new technique in file sequencing," Technical Report, Ottawa, Canada, 1966, IBM Ltd.
- [88] A. Gionis A, P. Indyk, and R. Motwani, Similarity Search in High Dimensions via Hashing, *Proc. Int'l Conf. on Very Large Data Bases*, 1999, pp. 518-529
- [89] N. Imielinski N and T. Imielinski, "GeoCast – Geographic Addressing and Routing," *Proc. IEEE/ACM MobiCom*, 1997, pp. 66-76.
- [90] G.G. Finn G.G, "Routing and addressing problems in large metropolitan-scale internetworks," *ISI Research Report*, ISU/RR-87-180, 1987.
- [91] E. Kranakis, H. Singh, J. Urrutia, "Compass routing on geometric networks," *Proc. CCCG*, 1999, pp. 51-54.
- [92] P. Bose, P. Morin, "Online routing in triangulations," *Proc. ISAAC*, 1999, pp. 113-122.
- [93] H. Takagi, L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Trans. Communication*, vol. 32, no.3 , 1984, pp. 246–57.

- [94] I. Stojmenovic and X. Lin, "Loop-Free Hybrid Single-Path/Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks," IEEE Trans. on Parallel and Distributed Systems, vol. 12, no. 10, 2001, pp. 1023-1031.
- [95] P. Bose, P. Morin, I. Stojmenovic, J. Urrutla., "Routing with guaranteed delivery in ad hoc wireless networks," Journal of Mobile Communication, Computation and Information, vol. 7, no. 6, 2001, pp. 48-55.
- [96] B. Karp and H.T. Kung, "Greedy Perimeter Stateless Routing for Wireless Networks," Proc. IEEE/ACM MobiCom, 2000, pp. 243-254
- [97] K. Seada and A. Helmy, "Efficient Geocasting with Perfect Delivery in Wireless Networks," Proc. IEEE WCNC 4, 2004, pp. 2551-2556.
- [98] F. Kuhn, R. Wattenhofer, Y. Zhang, A. Zollinger, "Geometric ad-hoc routing: of theory and practice," Proc. Symp. on Principles of distributed computing, 2003, pp.63-72.
- [99] J. Ledlie, P. Pietzuch, M. Mitzenmacher, and M. Seltzer, "Wired Geometric Routing," Proc. IPTPS, 2007.
- [100] A.E. Abdallah, T. Fevens, and J. Opatrny, "Randomized 3D Position-based Routing Algorithms for Ad-hoc Networks," Proc. MOBIQUITOUS, 2006, pp. 1-8.

Abstract :

Multi-participant virtual environments are network applications that simulate the experience of real-time interaction among multiple users distributed all over the internet. In order to share the common sense of time and place, multiple entities interacting in the shared virtual environment should become aware of the modification to the shared environment, or the result of on the other objects interactions shortly after such events occur. This is achieved through the distribution of update messages. As the number of users and entities increases, there is a considerable increase in the amount of update messages exchanged among the users and entities. This is considered as one of the most important challenges in the design of large scale virtual environments. The use of distributed architectures is a solution to this problem. However, such architectures should in turn deal with a variety of challenges related to noncentralized protocols.

This dissertation proposes a number of distributed architectures for massively multi user virtual environments in an attempt to provide an efficient infrastructure for update message distribution. Simulation has also been used to demonstrate the functionality and performance of the proposed approaches.

Keywords : Massively Multi User Virtual Environments (MMVEs) , Peer to Peer networks, Distributed algorithms, Ant colony optimization, Hilbert Curve, KD Tree, Wired Geometrical routing



**Sharif University of Technology
Faculty of Electrical Engineering**

Ph.D. Thesis

**Overlay routing Architectures for Massively Multiuser Virtual
Environments**

**By
Behnoosh Hariri**

**Supervisors:
Dr Mohammad Hossein Alavi
Dr Mohammad Reza Pakravan**

August 2009