Archive of SID

بررسی یک مدل جدید الهام گرفته از ساختار درختی در حراج اینترنتی برای افزایش مقیاس پذیری و کارایی سیستم

بهرنگ برکتین، دکتر محمد رضا خیام باشی
عضو هیات علمی گروه کامپیوتر دانشگاه آزاد واحد نجف آباد
Behrang_Barekatain@yahoo.com
عضو هیات علمی گروه کامپیوتر دانشکده فنی مهندسی دانشگاه اصفهان
M.R.Khayyambashi@eng.ui.ac.ir

۱-مقدمه:

هدف ما در این مقاله ارایه یک مدل مناسب با استفاده از توپولوژی Loop – star الهام گرفته از ساختار درختی جهت ارتباط سرورهای حراج اینترنتی با یکدیگر است و قصد داریم تا به کمک آن انجام یک حراج اینترنتی و در نهایت پیاده سازی پروسه مدیریت زنجیره تامین را با سهولت و دقت بیشتری امکانپذیر نماییم. همچنین قصد داریم نشان دهیم که این روش پیشنهادی کارایی لازم ومناسبی را نسبت به سایر مدلهای موجود دارا می باشد. کارا بودن این توپولوژی در پایان مقاله اشاره شده است و نشان می دهد که این توپولوژی پیشنهادی کارایی خوبی را در پاسخگویی به تقاضاهای مخدوم[1] ارایه می دهد.

در دنیای امروز مشتریان تمایل زیادی برای استفاده از فن آوری تجارت الکترونیکی دارند، بطوریکه یک مشتری راغب است تا با ورود مجازی به یک فروشگاه از کاتالوگهای الکترونیکی برای انتخاب کالای موردنظرش استفاده کرده و توسط یکی از روشهای پرداخت الکترونیکی مبلغ آن را پرداخت نماید [۱]. در این حین ارتباط معنی داری بین فروشنده وخریدار برقرار می شود، بطوریکه طرفین پس از تأیید یکدیگر و نوع روش پرداخت، از انجام معامله ای ساده و بی دردسر راضی وخوشحال خواهند بود بالاخص اگر از نوع حراج نیز باشد[۲].

حراج اینترنتی یکی از مهیج ترین ابزاری است که در اینترنت استفاده می شود. زیرا توسط آن همه افراد به سادگی می توانند کالاهای مورد نیاز خود را به ساده ترین و ارزانترین روش و در کمترین زمان ممکن ودر هر ساعتی از شبانه روز بدون نیاز به حضور فیزیکی در محل حراج خریداری نمایند. امروزه روشهای متعددی برای انجام یک حراج اینترنتی وجود دارد که مهمترین آنها عبارتند از :

- **روش English** : بسیاری از حراج ها در اینترنت از نوع English هستند[۳]. انجام حراج در روش English بدین صورت است که همه شرکت کنندگان سعی در ارایه بالاترین قیمت دارند. ابتدا فروش کالابا ارایه یک پیشنهاد پایه آغاز می شود و سپس شرکت کنندگان Bid های متفاوتی را برحسب توان خرید خود ارایه می کنند و در نهایت بالاترین قیمت بعنوان برنده حراج شناخته خواهد شد [۴].
- **Sealed bid**: در این روش خریداران پیشنهادات خود را در یک مهلت مقرر و معین ارایه می دهند و سپس این پیشنهادات بصورت محرمانه نزد یک Auctioneer می ماند تا زمانی که مهلت ارایه پیشنهادات به پایان برسد. در نهایت قیمتهای ارایه شده پس از پایان مهلت ارسال آنها بررسی شده و برنده مشخص می شود[3] .
- **Single Round Sealed Auction**: در این روش هیچگونه رقابتی وجود ندارد، بطوریکه برخلاف روش English که شرکت کنندگان را تشویق به دادن قیمت بالاتری نسبت به سایرین می نمود، در این جا چنین حالتی وجود ندارد. بعبارتی شرکت کنندگان پیشنهادات خود را همزمان ارایه می دهند و فروشنده بهترین پیشنهاد را انتخاب می کند[3].
- **Multi Round Sealed Auction**: در این متد همه افراد شرکت کننده باید در یک مهلت مقرر و از پیش تعیین شده پیشنهادات خود را ارایه کنند. در پایان مهلت، برنده تعیین شده وگاهاً قیمت پیشنهادی فرد برنده شده به اطلاع سایر

[1] Client

www.SID.ir

شرکت کنندگان نیز می رسد و سپس یک حراج تازه دیگری با مهلت خودش آغاز می گردد. روش عملکرد در این حالت شبیه نوع single است[3].

- **روش Dutch**: این نوع حراجی بیشتر در جاهایی مفید است که فروشنده باید کالاهای خود را در سریعترین زمان ممکن به فروش برساند (مانند کالاهای فاسد شدنی). در این روش برخلاف روش English، ابتدا برگزارکننده حراج، حراج را با بالاترین قیمت شروع می کند وسپس به تدریج بر حسب شرایط، قیمت پایه را کاهش می دهد. حراج زمانی خاتمه می یابد که مهلت از پیش تعیین شده برای حراج پایان یابد، تمام اقلام به فروش برسند و یا قیمت از حداقل قیمت مجاز پایین تر بیاید[3]. گاه ممکن است ترکیبی از سه حالت فوق نیز در نظر گرفته شود.

آنچه مسلم است تمامی روشهای موجود برای انجام یک حراج اینترنتی باید دارای خصوصیات زیر باشند:

- **Confidential level**: در یک حراج اینترنتی باید میزان و حد مجاز انتشار اطلاعات معین باشد. گاهی اوقات ما حتی تمایل نداریم که قیمت برنده حراج را اعلام کنیم.
- **Rules in Processing an Auction**: قوانینی که در شروع، خاتمه و پایان یک روش حراج اینترنتی وجود دارند باید کاملاً مشخص و از پیش تعیین شده باشند تا ضمن برگزاری حراج، جای هیچ شکی باقی نماند. این قوانین شامل نحوه ارتباط مخدوم ها با سرورها و همچنین خود سرورها با یکدیگر می شود.
- **Limits in an Auction**: اصولاً در هر حراجی یک قیمت اولیه و حداقل وجود دارد که قیمت های پیشنهادی نباید از آن کمتر شوند وگرنه حراج نخواهیم داشت و در واقع نوعی آتش زدن به اموال است. همچنین مهلت مقرر برای ارایه پیشنهادات و یا معتبر بودن یک پیشنهاد نیز باید قانونمند باشد. همچنین در انجام یک پروسه حراج اینترنتی چندین نقطه کلیدی وجود دارد که همه آنها در کنار یکدیگر پروسه انجام حراج را ممکن می کنند. این نقاط عبارتند از:
- **مخدوم**: منظور از مخدوم سیستمی است که شرکت کننده در حراج توسط آن قیمت پیشنهادی خود را اعلام می کند.
- **سرور**[1]: در یک حراج اینترنتی یک یا چند سرور وجود دارند که نقش اداره کننده حراج را بازی می کنند. در حقیقت تضمین اجرای قوانین حاکم بر حراج شامل سه خصوصیت بیان شده فوق الذکر بعهده این سرورها است. به طور خاص در مبحث ما به این سرورها Auction server می گویند.
- **قوانین**[2]: منظور از قوانین دستورالعمل ها و قوانینی است که نحوه تعامل بین مخدومها و سرورها و نیز خود سرورها با یکدیگر را بیان می کند.
- **ساختار ارتباطی بین عناصر**[3]: این بخش شامل مسیرها، تجهیزات سخت افزاری و نرم افزاری لازم، پروتکلهای ارتباطی و بطورخاص توپولوژی شبکه است.

با مروری بر موارد فوق در می یابیم که مهمترین مورد، ساختار ارتباطی بین عناصر است. زیرا اصلی ترین بحث در موارد اول و دوم سخت افزار می باشد که بیشتر از نقطه نظر هزینه قابل بررسی است. زیرا تمام سخت افزارها و نرم افزارهای امروزی به نحوی قابلیت اتصال به یکدیگر را دارند. قوانین نیز بستگی کامل به نوع روش پیاده سازی حراج دارند که آن هم باید توسط ارایه دهنده یا پیاده سازی کننده حراج تعیین گردد. اما آنچه بسیار مهم به نظر می رسد چگونگی ارتباط بین اجزای یک حراج اینترنتی از نقطه نظر ساختاری است. این مورد در قالب توپولوژی و چگونگی تعامل پروتکلی بین این اجزا قابل بررسی می باشد. هدف ما در این مقاله معرفی یک توپولوژی ساختیافته است که ضمن برخورداری از قابلیتهای قابل توجه، می توان از آن بعنوان یک ساختار مناسب شبکه ای در یک حراج اینترنتی استفاده نمود.

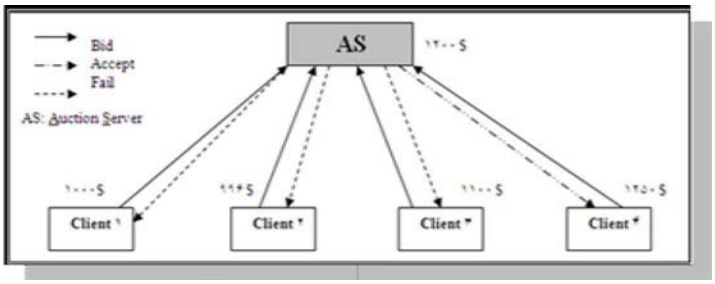
[1] Server
[2] Rules
[3]- شامل Server, Client و Path

www.SID.ir

۲- دیدگاه های مختلف در یک حراج اینترنتی:

در انجام یک حراج اینترنتی نقاط کلیدی وجود دارند که اگر به درستی مدیریت نشوند شاهد اعتراض و سردرگمی شرکت کنندگان خواهیم بود. انجام صحیح و سریع درخواست یا پیشنهاد یک مشتری و در نهایت تأیید یا رد آن پیشنهاد به نظر کار ساده ای می آید، اما ذاتاً دارای پیچیدگیهای فنی خاص است که مدیریت صحیح بر روی آن باعث رفع مشکلات و افزایش کارایی کل سیستم و در نهایت انجام یک حراج موفق و بدور از مسائل ناگوار خواهد شد. بنابر این لزوم داشتن یک بستر مطمئن و کارآمد از ضروریات تبادل اطلاعات در هر شبکه ای بالاخص جایی که مبادله پول صورت می پذیرد است. در یک حراج اینترنتی علاوه بر موارد فوق، مقایسه بین قیمت های پیشنهادی را نیز داریم که به نوبه خود تعهدات اخلاقی را نیز در پی دارد. دو دیدگاه کلی برای پیاده سازی بسترهای ارتباطی در یک حراج اینترنتی وجود دارد: متمرکز[1] و توزیع شده[2] [6].

۲-۱- دیدگاه متمرکز: در این روش کلیه شرکت کنندگان درخواستهای خود را به یک سرور مرکزی ارسال می کنند و این سرور مسئول نهایی مقایسه پیشنهادات و تأیید یا عدم تأیید آنها است. همچنین کنترل کلیه قوانین حاکم بر حراج نیز بعهده این سرور است. پیشنهاد دهنده قیمت خود را به سرور اعلام می کند. سپس سرور تقاضای رسیده را در صف قرار داده و بررسی می نماید. اگر پیشنهاد رسیده پایین تر از آخرین پیشنهاد ثبت شده باشد، به ارسال کننده آن یک پیام Fail می فرستد و در غیر این صورت با ارسال یک پیام Accept پیشنهاد دهنده را از پذیرفته شدن پیشنهادش آگاه می سازد. سرور پس از پذیرش آخرین پیشنهاد تا مدت معینی آن را معتبر می داند و اگر پس از پایان آن مهلت پیشنهاد بهتری دریافت نکرد، برنده را اعلام می کند و حراج پایان می پذیرد. نحوه انجام این روش در شکل یک آمده است.



شکل یک – نحوه پیاده سازی دیدگاه متمرکز

منظور از Bid همان تقاضای ارسالی از طرف مخدوم است. پیاده سازی این روش دارای مزایا و معایب خاص خود است که به برخی از آنها اشاره می کنیم. از جمله مزایای این دیدگاه عبارتند از:

- مدیریت آسان برروی پیشنهادات و بررسی آنها بدلیل متمرکز بودن پروسه مرکزی
- سادگی در طراحی ساختار و ارتباطات
- هزینه کمتر در پیاده سازی

برخی از مشکلات آن نیز عبارتند از:

- کاهش تحمل سیستم در برابر شکست های احتمالی[3] بدلیل احتمال بالای از کارافتادگی سرور و در نهایت شکست حراج. در واقع دراین روش فقط یک کنترل کننده مرکزی داریم که از کار افتادن آن معادل شکست کل سیستم است[4].
- تأخیر در پاسخگویی به پیشنهاد دهندگان بدلیل ازدیاد پیشنهادات بررسی نشده در صف سرور
- توسعه پذیری سخت است

[1] Central
[2] Distributed
[3] Fault – Tolerance
[4] Point Of Failure

www.SID.ir

- بی طرف[1] نبودن سیستم

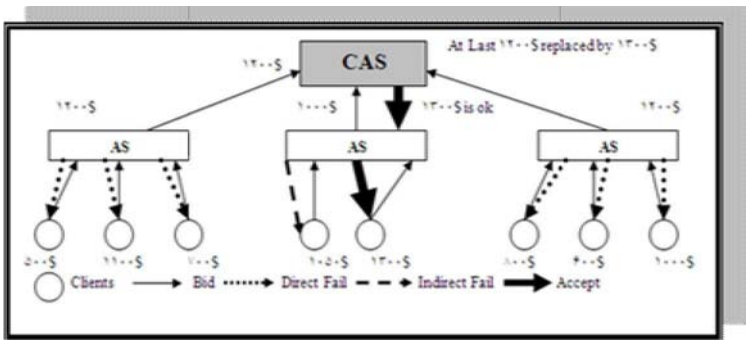
با توجه به مشکلات روش اول حرکت به سمت روش دیگری پیشنهاد می گردد که در آن چندین سرور در همکاری با یکدیگر اقدام به ارایه سرویس به مشتریان حاضر در یک حراج اینترنتی می نمایند. این حرکت منجر به پیدایش دیدگاهی موسوم به دیدگاه غیرمتمرکز یا توزیع شده گردید.

۲-۲- دیدگاه توزیع شده: در این روش ما از چندین سرور در کنار یکدیگر جهت پاسخگویی به تقاضاهای رسیده استفاده می کنیم. زمانی که یک مشتری پیشنهاد قیمت را ارایه می دهد، وظیفه سرور دریافت کننده این است که آن پیشنهاد را با بالاترین پیشنهادی که خود می داند مقایسه کند. اگر کمتر یا مساوی آن بود پس بطور طبیعی نمی تواند قابل قبول باشد. اما اگر قیمت پیشنهاد شده از بالاترین پیشنهادی که آن سرور می داند بالاتر بود باز هم سرور نمی تواند بطور قطع پیام Acceptای را به سمت پیشنهاد دهنده ارسال کند. زیرا ممکن است در سرور دیگری پیشنهادی بالاتر از این پیشنهاد ثبت شده باشد. نتیجه ای که می گیریم این است که اگر سرور بخواهد بطور مطمئن جواب پیشنهاد دهنده را بدهد باید بتواند با سایر سرورها و در نهایت با CAS[2] ارتباط داشته باشد تا مقایسه ای صحیح بین پیشنهادات به انجام برساند. منظور از CAS سروری است که بعنوان مدیر همه Auction server ها عمل کرده و در هر لحظه بهترین پیشنهاد را نگهداری می کند. شکل شماره دو چگونگی پیاده سازی این روش را نشان می دهد.

این روش نیز دارای محاسنی است که به نوعی برطرف کننده نواقص دیدگاه متمرکز است. این مزایا عبارتند از:

- در صورت از کار افتادن یک AS، کل سیستم دچار اختلال نمی شود.مخدوم نیز پس از زمان مشخصی درخواست خود را به یک AS دیگر در همان ستاره یا ستاره دیگری ارسال می کند.
- بدلیل این که پاسخگویی به پیشنهادات رسیده بین ASها تقسیم می شود، بنابر این دیگر شاهد ازدحام پیشنهادات در صف یک AS نیستیم.
- توسعه پذیری سیستم بسیار آسانتر خواهد بود.

اما مهمترین مشکلی که هنوز وجود دارد وجود یک نقطه شکست به نام CAS است.



شکل دو- نحوه پیاده سازی دیدگاه توزیع شده

به جز مشکل مذکور، این دیدگاه دارای نکات قابل بررسی دیگری نیز هست که اگر بتوانیم آنها را لحاظ کنیم یا بهبودی در آنها ایجاد نمائیم می توان گفت که به یک ساختار نسبتاً مطمئن و کاملی رسیده ایم که در آن پروسه انجام حراج اینترنتی بطور اصولی تر و با ریسک کمتری صورت می پذیرد. این نکات می تواند بصورت ذیل مطرح شود:

۱- با توجه به مطالب فوق دریافتیم که سرورها باید با یکدیگر در ارتباط باشند. بنابر این سیاست گذاری هایی برای نحوه ارتباط آنها با یکدیگر مورد نیاز است. این که سرورها چگونه با یکدیگر ارتباط داشته باشند تا بتوانند اطلاعاتشان را مبادله کنند

[1] Fairness
[2] Centre Auction Server

www.SID.ir

در قالب نحوه پیاده سازی توپولوژی شبکه قابل بررسی است. بنابر این یکی از مهمترین نکات، وجود یک توپولوژی ساخت یافته بین سرورها می باشد که در صورت عدم وجود، باعث بروز مشکلات پیچیده تری نسبت به روش متمرکز خواهد شد.

۲- نکته مهم دیگر هزینه ایجاد روشی است که در آن CAS با از کار افتادنش باعث شکست کل سیستم نشود.

حل مشکل دوم را در ابتدا با پیشنهاد یک Backup Server[1] که بعنوان پشتیبان CAS اطلاعات ثبت شده آن را نگهداری می کند تا حدی برطرف می کنیم. این پیشنهاد طی سالهای گذشته نیز بعنوان یکی از اصلی ترین راهکارها برای حل این معظل مطرح شده است. در واقع BS با انجام عملی به نام کپی[2]، اطلاعات خود را با CAS سازگار نگه می دارد. وجود سازگاری در انجام یک پروسه کپی از مهمترین نکات است.

این سازگاری[3] توسط وجود یک پروتکل مطمئن قابل دستیابی است. [۵] بطور کلی دو تکنیک برای پیاده سازی عمل کپی وجود دارد [7] که عبارتند از:

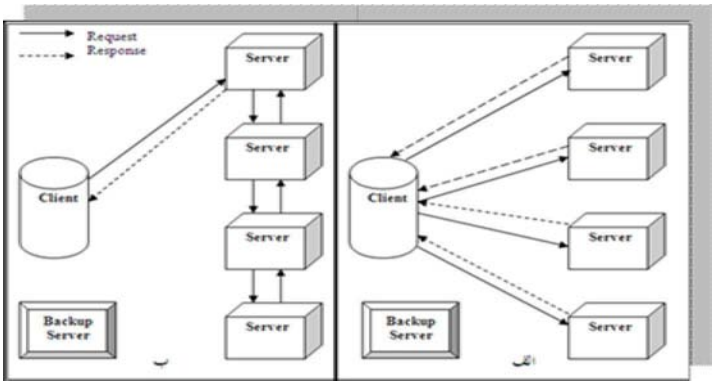
- **Active Replication**: در این روش تمامی سرورهایی که در پروسه کپی شرکت دارند باید تقاضای رسیده از یک مخدوم را پاسخ دهند. بعبارتی Replica ها مستقل از یکدیگر پاسخ داده و به نوعی روش همه– مطمئن[4] را پیاده سازی می کنند که در آن همه باید پاسخ بدهند. از محاسن این روش پایین بودن زمان پاسخ[5] است. اما این روش نیز همانند هر نوع پیاده سازی دارای معایبی نیز هست. از جمله این معایب می توان به این موارد اشاره نمود.

  i) منابع مورد نیاز برای پیاده سازی و انجام روش قابل ملاحظه است.

  ii) تقاضاها باید در یک ترتیب مشخص[6] انجام شوند.

  اصطلاح ترتیب مشخص بدین معنی است که نتیجه یک پروسه کاملاً وابسته به وضعیت اولیه ایستگاه سرویس دهنده بوده و ترتیب در اجرای تقاضاها بسیار مهم است. بعبارتی نتایج یک فرآیند وابستگی کامل به ورودیهای اولیه دارد. بنابر این سروهای چند نخی[7] معمولاً از حالت ترتیب غیر مشخص استفاده می کنند.

- **Passive Replication**: در این روش یکی از Replica ها بعنوان Primary عمل می کند. در واقع این سرور مسئول دریافت تقاضا از مخدوم ها و پاسخگویی به آنها است. Backup ها تنها با Primary مرتبط هستند و آخرین وضعیت موجود را از آن دریافت می کنند. در این روش مشکلات روش Active تا حد زیادی کم رنگ تر است. بعبارتی نیاز به منابع فراوان نبوده و هیچ تکیه ای بر وجود روش ترتیب مشخص برای پردازش تقاضاها نیست. شکل سه نحوه انجام این دو تکنیک را نشان می دهد.



[1] BS
[2] Replication
[3] Consistency
[4] All – Reliable
[5] Response Time
[6] Deterministic
[7] Multi Thread Servers

www.SID.ir

شکل سه- (الف) پیاده سازی تکنیک Active (ب) پیاده سازی تکنیک Passive

با توجه به مطالب مذکور ، یک توپولوژی مناسب جهت ارتباط خود از روش Passive Replication استفاده می کند و دیدگاه استفاده شده در آن دیدگاه توزیع شده است.

- در حقیقت استفاده از روش Passive بالاخص در محیط های توزیع شده مزایا و معایبی از جمله موارد زیر را به دنبال دارد:

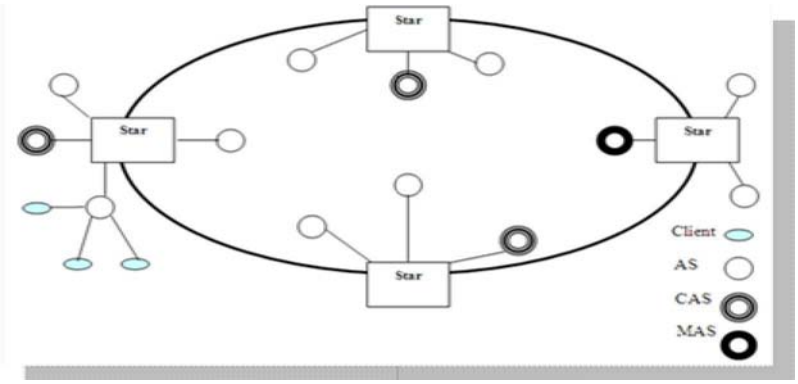
- بدلیل این که مخدوم مجبور نیست با تعداد زیادی سرور بطور همزمان در ارتباط باشد، بنابر این طراحی نرم افزارهای لازم درسمت مخدوم ساده تر خواهد بود و مخدوم نیز اجباری به درگیری کاری با چندین سیستم مختلف نخواهد داشت.

-بدلیل فواصل طولانی برخی سرورها با مخدوم مسائلی همچون انقضای مهلت[1] شدن دریافت Reply از سرورها در مخدوم پیش می آید که می تواند به شکست کل پروسه منجر شود.

-امکان ایجاد تغییر در روش کار در این حالت ساده تر است. زیرا اگر قرار باشد روش انجام کار تغییر کند، اعمال این تغییرات در یک سرور آسانتر از تعداد زیادی مخدوم است. منظور از روش انجام کار نحوه ارتباط مخدوم با سرورها است.

۳-معرفی مدل پیشنهادی در قالب توپولوژی Loop–Star:

با توجه به مطالب گفته شده تا به حال، به این نتیجه رسیدیم که استفاده از روش توزیع شده به همراه Passive Replication انتخاب مناسبی برای ما می باشد. شکل چهار ساختار کلی این توپولوژی پیشنهادی را نشان می دهد. دلیل استفاده از Star، قابلیت ذاتی بالای تحمل پذیری خطای آن است.



شکل چهار- ساختار کلی مدل پیشنهادی در سیستم

قبل از این که به بررسی این ساختار بپردازیم، لازم به ذکر است که هر ستاره بعنوان یک گروه دارای یک CAS می باشد که مسئول آن گروه است و یکی از CAS ها بعنوان MAS[2] نظارت بر کل CAS ها در کل ساختار سیستم که شامل همه ستاره ها می باشد را بعهده دارد.

در شکل چهار یک بستر حلقه ای موجود است که در آن هر ند خود یک شبکه با توپولوژی ستاره است. هر ستاره بعنوان یک گروه مطرح بوده و در هر گروه تعدادی AS قراردارد که یکی از آنها بعنوان CAS، سرگروه بقیه است. هر AS نیز تعدادی مخدوم را سرویس می دهد. منظور از سیستم مجموع کلیه عناصر و ارتباطات موجود در شکل چهار است. نحوه ارتباط بین مخدومها ، ASها و CASها بدین صورت است که یک مخدوم پیشنهاد خود را به AS متصل به خودش ارسال می کند. سپس این پیشنهاد در AS مورد بررسی قرار می گیرد و در صورتیکه از بالاترین پیشنهاد ثبت شده در AS پایین تر یا مساوی بود،

[1] Time out
[2] Main Auction Server

www.SID.ir

مخدوم از AS یک پیام Fail دریافت خواهد کرد. در غیر این صورت AS با ارسال تقاضایی از CAS، آخرین پیشنهاد ثبت شده در آن را دریافت می کند و پاسخ Accept یا Fail را به مخدوم ارسال می نماید.

در CAS نیز دو حالت بوجود می آید. اگر پیشنهادی که AS به آن داده پایین تر از آخرین پیشنهاد ثبت شده است که در این حالت AS نیز یک پیام Fail دریافت می کند و در نهایت این پیام به مخدوم می رسد. ممکن است این پیشنهاد بالاتر از آخرین پیشنهاد ثبت شده ای باشد که CAS دارد.CAS در این وضعیت تقاضای دریافت آخرین پیشنهاد ثبت شده در MAS را می کند. اگر این پیشنهاد مساوی یا پایین تر از آخرین پیشنهاد ثبت شده در MAS باشد، نهایتاً مخدوم یک پیام Fail از AS دریافت می کند ودر غیر این صورت پیام Accept به مخدوم می رسد. بدیهی است که دریافت هر پیام Accept ای از MAS به CAS و سپس به AS منجر به به روزرسانی آخرین وضعیت آنها می شود. همچنین واضح است که در صورت دریافت پیام Accept توسط یک مخدوم ممکن است تقاضای بهتری پس از آن در MAS ثبت شود که دراین صورت مخدوم جدید برنده فعلی حراج است.شکل(۵) شبه کد نحوه انجام این تراکنش را نشان می دهد:

```
Send Client.suggest     // Client send its suugest to AS
If Client.suggest <= AS.lastprice Then
    Send (AS.Client, Fail-Message)
Else
    Update AS.lastprice
    Send (CAS, Client.suggest)     //perhaps CAS's price ishigher than Client's su
 If Cleint.suggest <= CAS.lastprice Then
     Send (CAS.AS.Client, Fail-Message)
Else
     Update CAS.lastprice
     Send (MAS, Client.suggest) //perhaps MAS's price is higher than Client's suggest
If Client.suggest <= MAS.lastprice Then
     Send (MAS.CAS.AS.Client, Fail-Message)
Else
      Update MAS.lastprice
      Send (MAS.CAS.AS.Client, Accept-Message) //Cient's suggest is the highest
```

شکل پنج- شبه کد نحوه ارتباط مخدوم با AS و CAS

نحوه انتخاب CAS در یک گروه و در نهایت MAS کل سیستم یک مسئله مهم می باشد که می تواند باعث افزایش یا کاهش راندمان سیستم شود. بنابراین برای برگزاری یک همه پرسی جهت انتخاب CAS و MAS روشهای زیر پیشنهاد می گردد.

در یک گروه در ابتدا یکی از ASها بطور تصادفی یا بر اساس شماره اولویت اختصاص یافته بعنوان CAS گروه خود انتخاب می شود. سپس همه AS ها با این CAS مرتبط شده و آن را بعنوان سرگروه خود می پذیرند. مشکل زمانی بروز می کند که CAS فعال به هر دلیلی نتواند سرویس لازم را ارایه نماید. در این حالت باید انتخاباتی برگزار گردد. برای تمایز قائل شدن بین حالتی که CAS واقعاً خراب شده است و این که پاسخ از CAS دریافت نمی شود با توجه به ساختار پیشنهادی نیاز به انجام کار خاصی نیست. زیرا در هر دو حالت نتیجه نهایی عدم در دسترس بودن CAS خواهد بود. هر چند می توان با انجام تعداد مشخصی تلاش در واحد زمانی معین جهت دسترسی به CAS، بین این دو حالت تمایز قائل شد.در زیر به معرفی دو روش برای انتخاب CAS در یک ستاره می پردازیم.

۳-۱- روش Send Message Priority – Base:

www.SID.ir

در این روش هر AS دارای یک شماره اولویت[1] معین است که می تواند قابل تغییر باشد. این اولویت بر اساس پارامترهایی همچون قدرت سخت افزاری،نوع سیستم عامل و سایر مشخصات از این گونه قابل اختصاص به هر AS می باشد. اگر یک AS نیاز داشته باشد تا با CAS ستاره خود ارتباط برقرار نماید، پیغامی را به آن CAS ارسال و در یک بازه زمانی معین منتظر پاسخ می ماند. اگر پاسخی دریافت نکرد، پروسه همه پرسی را بدین صورت آغاز می کند. آن AS در ابتدا لیستی را به نام لیست اولویت[2] درحافظه خود ایجاد می نماید و شماره اولویت خودش را در آن قرار می دهد. سپس این لیست را درقالب یک پیام به نام ECAS[3] به تمام AS های گروه خود ارسال می نماید.

هر AS واقع در گروه یا همان ستاره، پس از دریافت یک پیام ECAS، ضمن ثبت شماره اولویت فرستنده پیام در لیست ایجاد شده در حافظه خود، شماره اولویت خود را نیز ثبت می‌نماید. در واقع هر AS ای به محض دریافت یک پیام ECAS اگر هنوز لیست اولویت خود را ایجاد نکرده است سریعاً آن را بوجود می آورد. سپس هر AS دریافت کننده پیام ECAS یک پیام مشابه آن، اما با درج شماره اولویت خودش به همه ارسال می کند. در نهایت پس از حداکثر 2T تمامی ASهای گروه دارای جدولی شامل همه شماره اولویت های ASهای گروه خود باضافه شماره اولویت خود هستند. عدد T زمان لازم برای ارسال یک پیام ECAS و دریافت آن در AS دیگر است. هر AS جهت اطمینان از دریافت کامل همه ECAS های ASهای دیگر گروه خود، می تواند به اندازه $2T+\alpha$ پس از ارسال ECAS خود صبر کند.[4] در نهایت AS جدول اولویت ایجاد شده خود را بررسی کرده و بالاترین اولویت را پیدا می کند. بطور قطع یکی از ASها که دارای اولویت بالاتری نسبت به سایر ASها است خود را بعنوان بالاترین اولویت می یابد. این AS به محض درک این موضوع سریعا خود را بعنوان CAS جدید معرفی نمی کند. در نهایت جهت اطمینان همه ندهایی که لیست خود را بررسی نموده اند، با ارسال پیامی به نام RECAS[5] به ASای که دارای بالاترین اولویت است کار بررسی خود را به اتمام می رسانند.ASای که در لیست خودش خود را بالاترین الویت یافته بود به محض دریافت حداقل یک RECAS نسبت به تأیید بالاتر بودن اولویت خودش مطمئن شده و با ارسال پیامی به نام SECAS[6] خود را بعنوان CAS جدید گروه معرفی می کند. توجه به نکات زیر در این روش ضروری است.

- هیچ دو ASای در یک گروه، اولویت یکسان ندارند.
- اگر CAS قدیم مجددا فعال شد با ارسال پیامی به CAS جدید شماره اولویت خود را با آن مقایسه می کند و در صورتیکه بالاتر بود خودش را بعنوان CAS جدید گروه به همه معرفی کرده و آخرین وضعیت را از CAS قبلی دریافت می کند.
- هر CAS جدید بعد از انتخاب توسط یکی از دو روش انتخاب MAS که در ادامه خواهیم دید، پروسه انتخاب MAS جدید سیستم را نیز اجرا می کند. زیرا ممکن است شرایط برای کسب عنوان MAS سیستم پس از فعال شدن مجددش برای آن فراهم شده باشد.
- هر AS جدیدی نیز که وارد یک گروه شود پروسه انتخاب CAS جدید را شروع می کند و اگر بعنوان CAS برگزیده شد در نهایت پروسه انتخاب MAS جدید سیستم را نیز انجام خواهد داد.

لازم به ذکر است که اگر پیامی از یک گروه به گروه دیگری برسد و هنوز CAS جدید آن گروه معین نشده یا CAS قبلی تازه فعال شده باشد، خود بعنوان CAS فعال نمی تواند پاسخگو باشد، زیرا آخرین وضعیت گروه را نمی داند. بنابر این در نهایت، مهلت دریافت پاسخ به مخدوم متقاضی انقضای مهلت شده و مجدداً تقاضای خود را ارسال می کند. این بار CAS جدید فعالیت عادی خود را شروع کرده است. همچنین با توجه به این که CAS جدید حتماً آخرین وضعیت ثبت شده اش کمتر یا مساوی CAS قبلی است، بنابر این هر گونه پیشنهاد بیشتر از آخرین وضعیت ثبت شده خود را پیگیری خواهد کرد و نهایتاً مخدوم پاسخ Fail یا Accept لازم را می گیرد.

---

[1] Priority Number
[2] Priority list
[3] Election CAS
[4] -$2T+\alpha$ شامل زمان رفت ECAS وبرگشت پاسخ آن می باشد که جهت اطمینان به اندازه $\alpha < T$ به آن اضافه می‌کنیم.
[5] Reply ECAS
[6] **Set ECAS**

www.SID.ir

حسن این روش این است که اگر هر ASای با هر اولویتی در گروه اضافه شود، هیچ نیازی نیست که بقیه ندها را از وجود آن آگاه کنیم. بلکه خود AS جدید با برگزاری الگوریتم انتخاب، ضمن معرفی خود به گروه ممکن است شانس دریافت عنوان CAS جدید گروه را نیز داشته باشد. این روش می تواند توسعه پذیری توپولوژی که یکی از نیازهای مهم در بسترهای مربوط به انجام حراج اینترنتی است را تضمین کند.شکل ۶ شبه کد مربوطه را نشان می دهد.

```
//When as AS creates, Administrator assigns to it a priority number which can change
I=0
Try n
   Send (AS,CAS,Request)      // AS send a request to its CAS
   Wait(AS,T)                 // AS wait for CAS's reply for T ms
    I=I+1
     If  I>n then    do priority.election(AS)      // if AS try for n times and didn't get reply
         call election procedure
 End try
 Send (AS.Client,Reply)       // AS get CAS's Fail/Accept reply and send it to Client

Procedure priority.election(ASj)
  Begin
    Create plist[j] for beginner ASj //AS which call this routine creates its priority list
    Plist[j].item=ASj.proirity-number   // ASj add its priority number to  its list
    Send (plist[j],ECAS,ALL)  //ASj send its list and ECAS(contain its name) to all AS in its STAR
   Do wait_RECAS(ASj,2*T + a)
   For each ASi in this STAR
       Get (ASi,ECAS,plist[j])
       If plist[i] not exist then
            Create plist[i]
       End if
       Copy (plist[i],plist[j])       // ASi copy ASj's list in its list
       Plist[i].item=ASi.proirity-number //ASi add its priority number to its list
       Send (plist[i],ECAS,ALL)   //ASi send its list to all other AS
       Do wait_RECAS(ASi,2*T +a)
    End for
End


Procedure wait_RECAS(ASk)
   Begin
      If  (2*T+a) is expired for ASk then
         Search (plist[k],highest_proirity,ASp) // search highest priority number in plist[k] and put its
                                                name into ASp
         If ASk=ASp and (ASk receives a RECAS from another AS) then
            Send (SECAS,ASp,ALL) // ASp send a SECAS to all
            which indicates that ASp is the new CAS in this STAR
  End
```

شکل شش –شبه کد مربوط به انتخاب CAS به روش Message Priority-base

۳-۲- روش Send Message Response – base:

www.SID.ir

نحوه انجام این روش نیز همانند روش قبلی است با این تفاوت که ASها به جای ارسال شماره اولویت خود تعداد تقاضاهای پاسخ داده شده ظرف مدت T[1] دقیقه گذشته را ارسال می کنند.ASای بعنوان CAS جدید معرفی می شود که دارای بیشترین تعداد پاسخ[2] ها در T دقیقه گذشته باشد. به نظر می رسد پارامتر Response پارامتر بهتری نسبت به شماره اولویت باشد. زیرا AS ای بعنوان CAS جدید معرفی می شود که بیشترین تعامل با مخدوم ها را داشته است. واضح است که اگر AS جدیدی به گروه وارد شود پروسه انتخاب را برگزار نخواهد کرد. زیرا عدد T برای آن صفر است یا بعبارتی تعداد پاسخ های آن هنوز صفر است . ولی اگر AS ای در گروه غیرفعال ومجدداً فعال شود (بالاخص اگر CAS بوده باشد) می تواند پروسه انتخاب CAS جدید را شروع کند. شبه کد این روش نیز همانند روش اول است. تنها به جای شماره اولویت تعداد تقاضاهای پاسخ داده شده توسط هر AS در لیست قرار می گیرد.

روشهای فوق جهت انتخاب یک CAS در یک گروه خاص بود. برای انتخاب MAS که در واقع مسئول اداره همه CASها و در نتیجه کل سیستم است و ارتباط Online ای با BS دارد روشهای دیگری را پیشنهاد می کنیم که شرح آنها بصورت زیر است. در مجموع برای گزینش یک MAS دو راهکار کلی پیشنهاد می گردد.

۳-۳-روش Token Ring Priority – base :

اگر در یک سیستم به این نتیجه برسیم که دیگر MAS فعال در دسترس نیست، اولین CASای که این مشکل را متوجه می شود با ارسال پیامی به سایر CAS های گروه های دیگر همه آنها را از این مسئله مطلع کرده و سیستم را به حالت بلوکه می برد. دراین وضعیت هیچ سرویسی به مخدوم ها داده نمی شود و طبیعتاً در سمت مخدوم ها انقضای مهلت هایی اتفاق می افتد که منجر به ارسال مجدد درخواست یا پیشنهاد آنها خواهد شد.

حالت بلوک شدن تا پایان انتخاب MAS جدید ادامه دارد. در این روش CASای که متوجه شکست سیستم در اثر خرابی MAS شده است، پروسه همه پرسی را آغاز می کند. این CAS با ساختن بسته ای به نام EMAS Token[3] اولویت و نام منحصر به فرد گروه خود را درون آن قرار می دهد. روش انتخاب شماره اولویت به همان صورت است که برای CAS عنوان شد.همچنین آدرس شبکه ای خود را نیز اضافه می نماید و توکن[4] را به سمت گروه مجاورش ارسال می نماید. در هر گروه نیز CAS آن گروه همین عملیات را انجام می دهد. اگر CAS ای در یک گروه توکنی را دریافت کرد که در آن مشخصات خود و گروهش قرار داشت، این بدین معنی است که توکن یک دور کامل در حلقه زده است. در این زمان CAS الویت های داخل توکن را بررسی و بالاترین شماره اولویت را می یابد. در این حالت با ارسال پیامی به آن CAS (اگر خودش نباشد) و همچنین پیامی به همه CAS های سیستم ضمن استخراج سیستم از حالت بلوکه، MAS جدید را نیز معرفی می کند. البته روش بهتر این است که خود MAS جدید با ارسال پیام ERMAS[5] به سایر CAS ها اطلاع دهد.

همچنین زمانی که توکن به گروه MAS قبلی می رسد دو حالت پیش می آید. (۱) در این گروه یک CAS جدید فعال شده که این CAS مشخصات خود را درون توکن می گذارد و آن را به گروه مجاورش ارسال می کند. (۲) هنوز در این گروه CAS انتخاب نشده است که در این وضعیت توکن بدون آن که چیزی به آن اضافه شود توسط یکی از ASها به گروه مجاور هدایت می شود. چرخش توکن در حلقه فقط در یک جهت است. شبه کد این روش در شکل شماره ۷ آمده است.

```
I=0
Try n
   Send (CAS,MAS,Request)   // CAS send a request to its MAS
   Wait(CAS,T)              // CAS wait for MAS's reply for T ms
    I=I+1
     If   I>n then
```

[1]- بر حسب شرایط شبکه قابل تعیین است.

[2] Response
[3] Election MAS Token
[4] Token
[5] Election Release MAS

www.SID.ir

```
        do priority.election(CAS)  // if CAS try for n times a didn't get reply call election procedure
End try
Send (CAS.AS.Client,Reply) //CAS get MAS's Fail/Accept reply and send it to Client

Procedure priority.election(CASj)
  Begin
    Send (Block_message,ALL)   // send message to all CAS to block system
   Create EMAS.Token(proirity,name)
   Add.Item.EMAS.Token(CASj.priority-number,CASj.name,CASj.address)
    Repeat
       Send (EMAS.Token, next_star,CASk) // send Token to next star's CAS
       Add.Item.EMAS.Token(CASk.priority-number,CASk.name,CASk.address)
    Until (CASj get EMAS token contains its priority-number and name)
    Search (EMAS.Token,highest_proirity,CASp) //put the Highest priority CAS name into CASp
   Send (ERMAS,CASp,ALL)  // CASj send a message to all other CAS which  indicates that
                                  CASp in the new  MAS
   send (Unblock_message,ALL) // CASj send a message to all to unblock the system
```

شکل هفت – شبه کد مربوط به انتخاب MAS به روش Token Ring Priority-base

۳-۴-روش Token Ring Response – base :

این روش نیز همانند روش قبلی است، ولی با این تفاوت که هر CAS به جای قرار دادن اولویت خود، تعداد Response های داده شده در T دقیقه گذشته خودش را درون توکن می گذارد و در نهایت CASای بعنوان MAS جدید شناخته می شود که عدد Response آن در T دقیقه گذشته بیشتر باشد. باز هم تأکید می شود که پارامتر تعداد Response ها در T دقیقه گذشته مناسبتر از شماره اولویت است. شبه کد این حالت همانند روش قبل است با این تفاوت که به جای شمار اولویت تعداد تقاضاهای پاسخ داده شده تا کنون قرار می گیرد. با توجه به مطالب فوق نحوه برقراری ارتباط بین ASها با CAS گروه خود و CASها با MAS سیستم می تواند بدین صورت باشد که تعدادی AS در یک گروه با یک CAS بعنوان مسئول گروه در ارتباطند و همه CAS ها با یک MAS که خود آن MAS بعنوان یک CAS در یک گروه مشغول به کار است مرتبط می باشند. در این دیدگاه هیچ ASای با MAS بطور مستقیم در ارتباط نیست، مگر این که آن MAS بعنوان CAS گروهی باشد که یک AS در آن گروه عضویت دارد. به هر حال از نظر منطقی همه ASهای این گروه، MAS را بعنوان CAS خود می شناسند. اصولا بدلیل این که خطر از کار افتادن MAS فعال سیستم وجود دارد، با دو مسئله زیر روبرو هستیم :

i) انتخاب یک MAS که الگوریتم آن را بررسی نمودیم.

ii) انتقال آخرین وضعیت سیستم به MAS جدید.

برای حل مشکل دوم پیشنهاد استفاده از یک Online Backup Server مطرح است که بطور دائم با MAS فعال در ارتباط می باشد. به محض این که MAS جدید فعال شد ارتباط خود را با BS برقرار نموده و آخرین وضعیت سیستم را از آن تقاضا می کند. سپس این وضعیت را با اطلاعات ثبت شده خود مقایسه نموده و اگر اطلاعات خودش به روزتر باشد آن را به BS انتقال می دهد[1] و همچنین اطلاعات BS را که در اختیار خود ندارد بر روی حافظه اش ثبت می کند. ارتباط بین MAS و BS کاملاً Online است. هر چند Online بودن هزینه بیشتری نسبت به Offline دارد، ولی باتوجه به حجم نه چند زیاد اطلاعات در این گونه سیستم ها مفید به نظر می رسد. لازم به ذکر است که BS می تواند وضعیت CASهای گروه ها را نیز ثبت کرده و در موقع لزوم به CAS جدید آن گروه ارسال نماید.

[1] زیرا ممکن است MAS جدید پیشنهادی داشته باشد که هنوز در BS ثبت نشده است.

www.SID.ir

نحوه تعامل مخدوم با ساختار سیستم و سرور ها باید به گونه ای باشد که مشتری احساس برخورد با یک سیستم پیچیده را نکند و در اسرع وقت پاسخ بگیرد. بعبارتی از کار افتادن یکی از سرورها در سمت مخدوم احساس نشود. با توجه به ساختار توزیع شده در توپولوژی پیشنهادی ، تمامی درخواستها به یک سیستم ختم نمی شود. بلکه در سه سطح AS , CAS و MAS مورد بررسی قرار می گیرد ، بطوریکه در هر سطح همه درخواستهای رسیده به سطوح بالاتر منتقل نمی شود. در ضمن به علت وجود تعداد زیادی ستاره که داخل آنها نیز تعدادی AS وجود دارد تقریبا بار ترافیکی شبکه در حالت بار بسیار سنگین، نسبت به سایر توپولوژیهای موجود به شدت کاهش می یابد.در این ارتباط اگر فرض کنیم :

N: تعداد کل درخواستهای مجموع همه مخدوم ها

P : تعداد Star های موجود در ساختار شبکه

M: متوسط تعداد AS ها در هر Star

باشد، آنگاه تعداد درخواستهایی که هر ستاره می گیرد U= N/P و تعداد درخواستهایی که هر AS جواب می دهد R=U/M خواهد بود. این روش پیشنهادی نشان می دهد که چگونه بار سیستم شکسته می شود. در ضمن از این R تقاضا برای هر AS تعداد Z تقاضا (Z<<R) به CAS و تعداد G تقاضا (G<<Z) به MAS منتقل خواهد شد. بنابراین تمامی تلاش بر آن است که پاسخ رد یا قبول پیشنهاد کاربر در همان AS داده شود. با توجه به مطالب فوق، قطعا سیستم دارای خروجی[1] بالا و زمان پاسخ پایینی[2] خواهد بود.

ساختار سیستم به گونه ای است که هرکاربر به راحتی می تواند به نزدیکترین ستاره متصل شود و دیگر مسافت عامل پیچیده ای در سیستم محسوب نخواهد شد و در نهایت Time Out مشکل پیچیده ای نخواهد بود. کاربر یا همان مخدوم در سریعترین زمان پاسخ خود را از AS می گیرد. اگر AS هم نتواند جواب دهد، تنها تقاضاهائی به بالاتر می رود که AS قیمت پیشنهادی آنها را بالاتر از خود ببیند. بنابراین ترافیک شبکه به شدت کاهش یافته و سرعت پاسخگوئی بالا می رود و در نهایت سیستم از نظر کاربر ایده آل به نظر می رسد. با توجه به دلایل زیر، مدل پیشنهادی کاملا نشان دهنده این موضوع است که با شکست هر عنصر سیستم حالت پایدار خود را حفظ می کند.

i) اگر AS دچار مشکل شود بلافاصله کاربر می تواند با AS دیگری در همان ستاره ارتباط برقرار کند و الزامی به ارتباط با AS درستاره دیگری نیست.

ii) اگر CAS یا MAS دچار مشکل شوند با الگوریتمهای انتخابی مطرح شده در این مقاله و در نهایت وجود BS سیستم حالت پایدار خود را بدست آورده و حفظ می کند.

همچنین به دلیل وجود ساختار ستاره در توپولوژی پیشنهادی امکان افزودن هر AS به سیستم به سادگی امکانپذیر است. همچنین قابلیت اضافه کردن Star نیز وجود دارد. این قابلیت یک امتیاز ذاتی برای توپولوژی های نوع ستاره به شمار می رود که میتوان به سادگی آنها را ایجاد و یا توسعه داد.

۴ – مقایسه توپولوژی جدید با توپولوژی های مرسوم و سنتی:

یکی از معروفترین و رایجترین توپولوژیهای موجود ساختار Tree یا درختی است که در این جا ابتدا آن را با توپولوژی پیشنهادی مقایسه می کنیم و سپس به بررسی توپولوژی پیشنهادی با سایر روشهای دیگر می پردازیم. لازم است در مقایسه این دو روش به موارد زیر دقت کنیم :

[1] Throughput
[2] Response Time

www.SID.ir

در توپولوژی پیشنهادی حداکثر تعداد سطوحی که یک مخدوم باید طی کند تا به یک MAS برسد و بتواند Bid خود را در آنجا مقایسه نماید مساوی سه سطح می باشد. یعنی حالتی که مخدوم در ستاره ای باشد که MAS در آنجا نیست. در این حالت مخدوم تقاضای خود را بصورت زیر دنبال می کند :

- ارسال Bid به AS متصل به خودش
- ارسال Bid از AS به CAS ستاره خودش
- ارسال Bid از CAS به MAS در ستاره دیگر

توجه داشته باشیم که در ساختار درختی، MAS همیشه در ریشه است. بنابراین اگر فرض کنیم تعداد یالهائی که باید طی شود تا به ریشه برسیم مساوی عدد سطح درخت باشد، آنگاه مخدوم باید به اندازه سطح درخت باضافه یک (فاصله تا AS خودش) را بگذراند تا به MAS برسد. بنابراین حداکثر تعداد سطوح قابل گذر بر خلاف روش این مقاله که بدون توجه به تعداد AS ها یا افزایش تعداد آنها به عدد سه محدود بود، در ساختار درختی به تعداد سطوح درخت بستگی مستقیم داشته و با افزایش آن زیاد خواهد شد. حال اگر فرض کنیم فاصله سیستمها از یکدیگر مهم نباشد و تعداد یال ارتباطی را بعنوان یک معیار اندازه گیری در نظر بگیریم، آنگاه برای توپولوژی پیشنهادی این مقاله حداکثر فاصله ارتباطی برای مقایسه یک Bid در یک رفت و برگشت از مخدوم به MAS سیستم مساوی شش (۳ * ۲ - با توجه به دنباله تقاضای مخدوم که در بالا مطرح شد) خواهد بود. این عدد با افزایش تعداد AS ها به هیچ عنوان زیاد نخواهد شد. زیرا با اضافه شدن یک AS، این سیستم یا در یک ستاره موجود جای می گیرد و یا در داخل یک ستاره جدید قرار داده می شود که در هر دو صورت تعداد سطوح زیاد نخواهد شد. بنابراین ساختار درختی تنها در حالتیکه دارای فقط یک سطح باشد می تواند بهتر از ساختار Loop-Star باشد.زیرا در این حالت طول ارتباط مساوی چهار خواهد بود(مخدوم به AS ، AS به ریشه و بالعکس این مسیر) . اما اگر تعداد سطوح درخت از دو بیشتر شود، آنگاه در همه حال طول ارتباطی این روش مناسبتر است. در حالتیکه تعداد سطوح درخت دو باشد، طول ارتباطی هر دو روش یکسان است. در واقعیت تعداد AS ها بسیار زیاد است. برای مثال با ۲۰۰ عدد AS، تعداد سطوح در یک درخت باینری کاملا متوازن (حالت کاملا متوزان، خوشبینانه ترین حالت درخت برای کاهش تعداد سطوح درخت است) مساوی هفت است که در نتیجه طول ارتباطی مخدوم تا MAS مساوی 16 = 2 * (7+1) خواهد بود. توجه داشته باشیم که در این مقایسه خوشبینانه ترین حالت درخت و بدبینانه ترین حالت توپولوژی پیشنهادی (یعنی حالتی که MAS در ستاره ای نیست که مخدوم در آن قرار دارد) را بررسی کردیم.تنها نکته ای که باقی می ماند این است که فرض کنیم درختمان را از نوع K تائی یک یا دو سطحی در نظر بگیریم. یعنی با اضافه شدن هر AS آن را بعنوان یک ند در سطح یک یا دو در درخت قرار دهیم و کاری کنیم که هیچگاه تعداد سطوح درخت از دو بیشتر نشود. بدیهی است که در این حالت طول ارتباطی همیشه مساوی یا بهتر از روش این مقاله می باشد. اما ساده ترین مشکل که در این گونه ساختن درخت بوجود می آید آن است که اگر ریشه از بین برود، ما دارای زیر درختهای بسیار زیاد، ولی با تعداد ندهای بسیار کم هستیم که در عمل هیچ کارایی مفیدی نخواهند داشت. زیرا اولا: زیر درختها هیچ ارتباطی با یکدیگر ندارند، ثانیا: بدلیل کم بودن تعداد AS ها به جرات می توان گفت که هر AS با احتمال بسیار پایین آخرین قیمت را نزد خود دارد. بدیهی است هر چه بیشتر به سمت درخت باینری پیش برویم، آنگاه با دادن یک Bid مناسب توسط یک مخدوم، تعداد AS های بیشتری تا ریشه به روز خواهند شد و در نتیجه اگر ریشه از کار بیفتد ما دارای دو زیر دخت هستیم که در آنها ASهائی وجود دارند که با احتمال بالاتری دارای آخرین Bidهای پیشنهادی هستند. در حالتی که درخت دارای یک سطح باشد، آنگاه با از بین رفتن ریشه دارای N عدد AS مستقل هستیم که هیچ ارتباطی با یکدیگر ندارند. یعنی افراز یک درخت N ندی به دقیقا N ند جداگانه که عملا به معنای از بین رفتن کل ساختار درخت و در نتیجه کل توپولوژی ارتباطی است.

توپولوژی Loop-Star مطرح شده در این مقاله میتواند نقاط متمایز کننده و قابل مقایسه ای نیز با روشهای مرسوم و سنتی ذیل داشته باشد :

www.SID.ir

۱- **ساختار سرور مرکزی** :همانطور که در ابتدای این مقاله عنوان شد در صورت داشتن فقط یک AS و افزایش زیاد مخدوم ها، ترافیک در صف تقاضای سرور زیاد خواهد شد و پس از مدتی حالت سرریز[1] در سرور یا انقضای مهلت در سمت مخدوم رخ خواهد داد. اما در این توپولوژی پیشنهادی طبق مباحث فوق بار بین AS ها تقسیم شده و مشکلات فوق تقریبا کم رنگ می باشد.

۲- **ساختار ستاره[2] محض** : هر چند ساختار ستاره خود باعث افزایش کارایی می باشد، اما زمانی که فقط یک ستاره داشته باشیم و فاصله ارتباطی AS با مخدوم یا حتی دو AS در یک ستاره زیاد شود، مشکل انقضای مهلت باز هم پابرجا خواهد ماند. ولی در این توپولوژی هر مخدوم به نزدیکترین ستاره مجاور خود متصل می شود و هر AS نیز عضو نزدیکترین ستاره خواهد شد. ساختار حلقه ای که ستاره ها را به هم متصل می کند اجازه داشتن چندین ستاره متصل به هم را می دهد و در صورتیکه در یک ناحیه خاص تعداد سیستمها زیاد شود، به سادگی می توان با تشکیل چند ستاره توسعه شبکه را ممکن و بار را بین سرورها تقسیم نمود. در این حالت توسعه پذیری شبکه نیز ساده تر به انجام می رسد.

۳- **ساختار حلقه محض** : وجود یک حلقه می تواند باعث اتصال چندین سیستم به یکدیگر شود. اما واضح است که افزایش تعداد سیستمها در یک ساختار حلقه ای محض کاملا محدود بوده و با افزایش بیش از حد سیستمها در حلقه کارایی شبکه پایین خواهد آمد. این مسئله کاملا از ساختار شبکه های Token Ring مبتنی بر حلقه مشخص است.

۵-نتیجه گیری

در این نوشتار ضمن بررسی توپولوژی موجود در حراجهای اینترنتی، مدلی جدید در قالب توپولوژی Loop-Star مطرح و به تفضیل بررسی گردید. این مدل پیشنهادی، ساختار جدیدی را در توپولوژی حراج اینترنتی ایجاد نموده است و مزایای عمده ای مانند توسعه پذیری آسان[3]، رعایت اصل بی طرفی[4] و توان عملیاتی[5] بالا را در بر دارد که آن را می تواند بعنوان یک توپولوژی موثر در مباحث حراج اینترنتی و در نهایت مدیریت زنجیره تامین قرار دهد.

[1] Overflow
[2] Star
[3] Scalability
[4] Fairness
[5] High Throughput

www.SID.ir

مراجع :

[1] F.Panzieri and S.K.Shrivastava, “on the prouision of Replicated Internet Auction Services.

[2] "تجارت الکترونیک" ماهنامه تخصصی دانش و کامپیوترسال هفتم،صفحه۴۲-۴۵ شماره ۵۹،نوشته بهرنگ برکتین-آذر۸۵

[3] M. Kumar and S.I Feldman. “Internet Auction.” Proc. of the 3rd USENIX workshop on Electronic Commerce, pp.44-60, 1998.

[4] Makoto Yokoo. “Internet Auctions: Theory and Application.” Journal of Japanese Society for Artificial Intelligence, Vol. 15, No.3, pp. 404-411, 2000.

[5] R.Guerraoui and A.Schiper, “Fault – tolerance: From Replication Techniques to group communication, “IEEE computer, vol. 30, pp. 68 – 74. Apr. 1997.

[6] P Ezhilchelvan,G. Morgan – International Symposium on Autonomous Decentralize Systems,2001.

[7] K PARK – Lecture notes in computer science – Springer

www.SID.ir