استفاده از تکنیک سطل‌بندی جهت ساخت درخت چندپخشی در سطح کاربرد شبکه

رضا بشارتی[۱]، مظفر بگ‌محمدی[۲]، ماشاءالله عباسی دزفولی[۳]
۱- کارشناس ارشد کامپیوتر، دانشگاه علوم و تحقیقات اهواز، r_besharati@yahoo.com
۲- استادیار، گروه کامپیوتر، دانشکده فنی و مهندسی، دانشگاه ایلام، mozafarb@ece.ut.ac.ir
۳- استاد، گروه کامپیوتر، دانشگاه علوم و تحقیقات اهواز، abbasi_masha@yahoo.com

چکیده

با توجه به مشکل پیاده‌سازی IP multicast در مسیریاب‌ها، درسال‌های اخیر روش‌های جایگزین زیادی در حوزه Application Layer multicast (ALM) به‌وجود آمده‌اند. در این مقاله ما با استفاده از تکنیک سطل‌بندی تمام گره‌های نزدیک به همدیگر را با کمک تعداد ثابتی نشانه که گره‌های شناخته شده‌ای می‌باشند در یک سطل قرار می‌دهیم. بعد از قرارگرفتن گره‌های نزدیک به همدیگر در یک سطل، از یک ساختار درختی برای ارتباط آنها استفاده می‌نماییم. با استفاده از یک تابع امتیازدهی که براساس ماندگاری هر گره به آن امتیاز می‌دهد یک مسؤل برای سطل مربوطه انتخاب می‌نماییم. هر مسؤل به‌عنوان ریشه زیر درخت انتخاب می‌گردد. با توجه به مجاورت گره‌های هر سطل با همدیگر، تأخیر زیردرخت ساخته شده در هر سطل کمتر از روش‌های مشابه می‌باشد. ازطرفی مجاورت گره‌ها باعث می‌شود زیردرخت مذکور دارای تنش زیادی نباشد. لذا درخت نهایی دارای تأخیر و تنش کمی است. از آنجایی‌که گره‌های ماندگارتر به‌عنوان مسؤل درنظر گرفته می‌شوند می‌توان گفت که این روش نسبت به روش‌های دیگر دارای تحمل‌پذیری بیشتر در برابر خطا (ترک گره) می‌باشد.

واژه‌های کلیدی

سطل‌بندی، تحمل‌پذیری خطا، ماندگاری، چندپخشی، میزبان، تنش، شکاف.

۱– مقدمه

چندپخشی[۱] یک روش کارآمد برای ارسال بسته‌ها از یک فرستنده به چندین دریافت‌کننده می‌باشد و باعث حذف بسته‌های تکراری در شبکه خواهد شد. از طرف دیگر با وجود چندپخشی، فرستنده نیازی به نگهداری اطلاعات مربوط به تمام گره‌ها ندارد [۱۶]، پس در یک شبکه باید امکان انجام عمل چندپخشی وجود داشته باشد. در IP multicast کار چندپخشی توسط مسیریاب صورت می‌پذیرد و تمام مسیریاب‌ها باید امکان انجام چنین عملی را داشته باشند. در شبکه‌های با مقیاس زیاد، مدیریت حالت‌های مربوط به گروه و پیاده‌سازی چند پخشی در مسیریاب‌ها غیرممکن به‌نظر می‌رسد. وجود مشکل پیاده‌سازی فوق‌الذکر باعث شده است که پیاده‌سازی چند پخشی در لایه کاربرد، یا به اصطلاح همان ALM مورد توجه قرار گیرد. روش‌های موجود برای ALM [۷ - ۱] زیر ساخت‌های شبکه را تغییر نمی‌دهند و برخلاف IP multicast که داده‌ها را در مسیریاب‌های انشعاب تکرار می‌کند، باعث تکرار داده‌ها در میزبان‌ها می‌شوند.

در روش‌های موجود برای ALM مشکلاتی وجود دارد که

Archive of SID

www.SID.ir

می‌توان به موارد زیر اشاره نمود. در ALM، میزبان‌ها عمل کپی و ارسال بسته‌های چندپخشی را انجام می‌دهند. لذا ممکن است یک بسته یکسان چند بار از یک لینک عبور کند. تعداد بسته‌های یکسان عبوری از یک لینک به‌عنوان تنش[۲] آن لینک تعریف می‌گردد. در IP multicast در تمام لینک‌ها تنش برابر با یک می‌باشد. در چندپخشی، رفتار میزبان‌ها از لحاظ پیوستن به گروه چندپخشی (یا ترک آن) کاملاً پویا می‌باشد. درصورت ترک یک یا چند میزبان غیربرگ، درخت ALM به زیردرخت‌های مختلف شکسته می‌شود. درنتیجه، درخت باید دوباره بازسازی شود که خود باعث تأخیر در ارسال داده خواهد شد. در بعضی از روش‌های موجود برای بالا بردن تحمل‌پذیری در برابر خطا (خرابی/ ترک‌گره)، تعدادی لینک اضافی بین اعضاء گروه نگهداری می‌شود و هر عضو اطلاعات زیادی در مورد بقیه اعضاء نگهداری می‌کند. این مسئله مقیاس‌پذیری گروه را کاهش می‌دهد.

در روش پیشنهادی (Bincast)، با استفاده از تکنیک سطل‌بندی[۳] تمام گره‌های نزدیک به همدیگر با کمک تعداد ثابتی نشانه[۴] که گره‌های شناخته شده‌ای می‌باشند در یک سطل قرار می‌گیرند. بعد از قرارگرفتن گره‌های نزدیک به همدیگر در یک سطل، از یک ساختار درختی برای ارتباط آنها استفاده می‌نماییم. به کمک یک تابع امتیازدهی که براساس ماندگاری هر گره به آن امتیاز می‌دهد یک مسؤل برای سطل مربوطه انتخاب می‌شود. این مسئول به‌عنوان ریشه زیردرخت انتخاب می‌گردد. با توجه به مجاورت گره‌های هر سطل تأخیر زیر درخت ساخته شده کمتر از روش‌های دیگر می‌باشد. ازطرفی مجاورت گره‌ها باعث می‌شود که زیردرخت مذکور دارای تنش زیادی نباشد. لذا درخت نهایی دارای تأخیر و تنش کمی است. از آنجایی‌که گره‌های ماندگارتر به‌عنوان مسئول درنظر گرفته می‌شوند می‌توان گفت که این روش نسبت‌به روش‌های دیگر دارای تحمل‌پذیری بیشتری در برابر خطا (ترک‌گره) می‌باشد. نهایتاً با توجه به این‌که از روش سطل‌بندی استفاده می‌نماییم و در هر سطل تعداد زیادی گره می‌تواند قرار گیرد مقیاس‌پذیری روش ارائه شده زیاد می‌باشد.

در ادامه در بخش دوم کارهای مرتبط عنوان شده است. در بخش سوم روش پیشنهادی به‌صورت کامل شرح داده شده است و نحوه اتصال گره جدید به گروه و انجام عمل شکاف[۵] مطرح می‌شود. در بخش چهارم نتایج شبیه‌سازی و در بخش پنجم نتیجه‌گیری ارائه شده است.

۲– کارهای مرتبط

روش‌های مختلفی برای ALM وجود دارد. در [۱۷] روش‌های موجود به دو دسته پروتکل‌های براساس درخت [۴ - ۲] و پروتکل‌های براساس مش [۱] و [۷] تقسیم شده‌اند. چند نمونه از این روش‌ها را بررسی می‌کنیم. درHMTP [۳] از یک ساختار درختی برای اتصال میزبان‌های شرکت‌کننده در گروه استفاده می‌شود. در این روش با آنکه تنش در لینک‌های موجود حداقل می‌باشد اما از لحاظ تحمل خطا (ترک یک یا چندگره) دارای تحمل‌پذیری کمی می‌باشد و با حذف (ترک) یک گره غیربرگ درخت مورد نظر به زیر درخت‌های مختلف تقسیم می‌شود و باید دوباره بازسازی گردد.

در NARADA [۱] بین اعضاء شرکت‌کننده در گروه یک ساختار مش شکل می‌گیرد. لینک‌های زیادی بین اعضاء گروه وجود دارد و هر گره لیستی از گره‌های گروه را نگهداری می‌نماید. این روش در برابر خطا (ترک یک یا چندگره) دارای تحمل بیشتری می‌باشد، اما از لحاظ مقیاس‌پذیری مشکل دارد. چون که هر گره باید لیست تعداد زیادی از اعضاء گروه را نگهداری نماید. از طرف دیگر این روش دارای تنش زیادی می‌باشد چون که لینک‌های زیادی بین اعضا وجود دارد [۲].

در NICE [۲] از یک ساختار سلسله مراتبی استفاده می‌شود و گره‌ها با کمک این ساختار در کنار یکدیگر قرار می‌گیرند. در سطح صفر گره‌های نزدیک به همدیگر در کلاسترهای مجزا با اندازه مشخص قرار می‌گیرند. اگر تعداد گره‌های قرار گرفته در یک کلاستر خیلی کم باشد، ممکن است که فاصله گره‌ها از همدیگر بسیار زیاد باشد. از طرف دیگر با بالا رفتن تعداد گره‌های موجود در کلاستر تنش نیز افزایش می‌یابد. در انتخاب مسؤل فقط موقعیت فیزیکی آن گره درنظر گرفته می‌شود و گره وسط کلاستر (از لحاظ فاصله) انتخاب می‌گردد. در سطوح بالاتر مسؤل‌های کلاسترها به‌صورت درخت با همدیگر در ارتباط می‌باشند. هر گره جدیدالورود باید از بالاترین سطح شروع به جستجو و Ping کردن نماید تا کلاستر مربوط به خود را پیدا کند. لذا پروسه ورود یک نود دارای مرتبه اجرایی $m \log n$ می‌باشد (m تعداد گره‌های هر کلاستر و n عمق درخت است).

سیستم‌های نظیربه‌نظیر(p2p) همانند Pastry [۱۱] و Tapestry [۱۰] برای ساخت overlay و مسیریابی، به‌صورت ضمنی از اطلاعات توپولوژی شبکه استفاده می‌کنند. این سیستم‌ها از ALM پشتیبانی کرده و فرض می‌کنند که توپولوژی مورد نظر دارای خاصیت "نابرابری مثلثی" است و از آن برای پیدا کردن گره‌های نزدیک در شبکه استفاده می‌کنند. اما مرجع [۵] ثابت کرد که در اینترنت "نابرابری مثلثی" همیشه برقرار نیست. سیستم‌های نظیربه‌نظیر t-CAN [۱۲] و eCAN [۱۴] به‌صورت صریح از اطلاعات توپولوژی استفاده می‌کنند. T - CAN فضای مورد نظر را همانند یک فضای کارتزین درنظر می‌گیرد و به هر گره یک محدوده ار این فضا را نسبت می‌دهد و از ترتیب نشانه‌ها برای کلاستر کردن گره‌ها موجود در شبکه استفاده می‌کند. eCAN با استمداد از

www.SID.ir

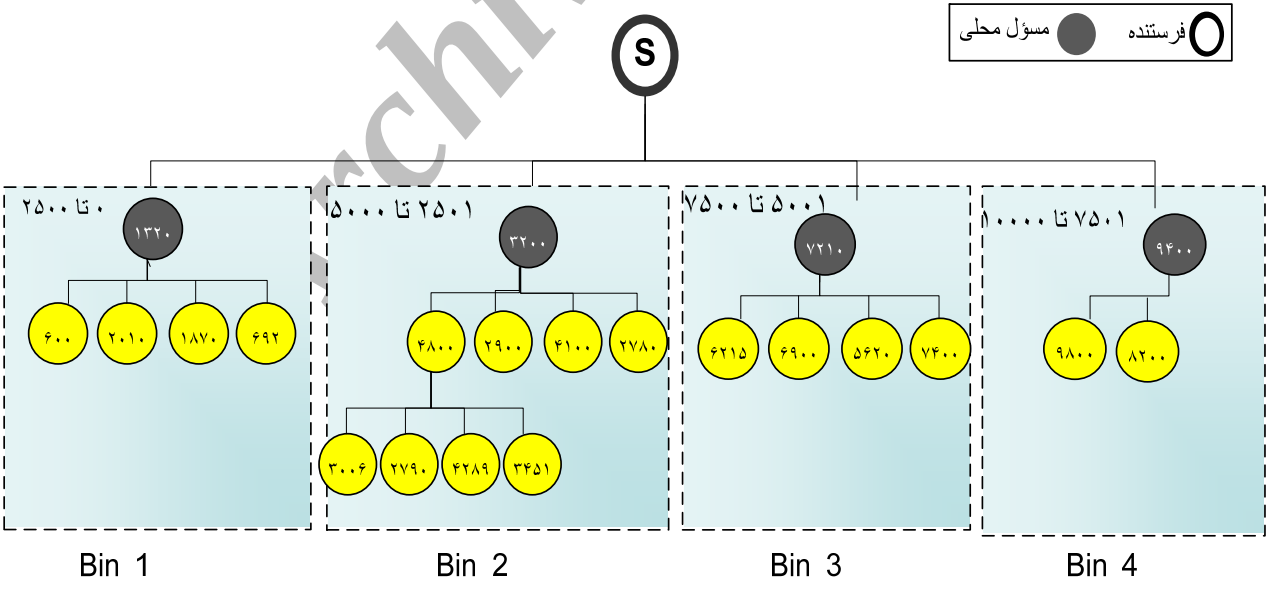
نشانه‌ها، یک شبکه overlay تحت عنوان expressway، برای بهبود مسیریابی در CAN می‌سازد. Brocade [۱۳] یک شبکه overlay بر روی Tapestry می‌سازد و از خاصیت "محلی بودن[۶]" مسیریابی در سطح AS استفاده می‌کند. Tag [۷] از هم‌پوشانی مسیر فرستنده در گیرنده‌های مختلف برای ساخت overlay استفاده می‌کند. Laptop [۱۵] از مکان جغرافیای نودها استفاده کرده و یک شبکه سلسله مراتبی را می‌سازد. در این الگوریتم هر گره موقعیت خود را با Ping کردن تعداد کوچکی از گره‌های موجود در شبکه بدست می‌آورد. الگوریتم مطرح شده در [۸] از مفهوم نشانه[۷] برای کلاستر کردن گره‌های موجود در شبکه استفاده می‌کند درحالی‌که الگوریتم موجود در [۹] از DNS برای کلاستر کردن استفاده می‌کند و گره‌های را که از لحاظ DNS به یکدیگر نزدیک باشند در یک کلاستر قرار می‌دهد. در [۱۶] از یکسری نشانه برای کلاستر کردن استفاده می‌شود در این روش هر گره تعدادی نشانه را Ping می‌کند و براساس ترتیب دریافت جواب از آنها کار کلاستر کردن را انجام می‌دهد و گره‌های که دارای یک ترتیب یکسان باشند را در یک کلاستر قرار می‌دهد.

۳– روش پیشنهادی

ابتدا مکانیسم سطل‌بندی را توضیح می‌دهیم. برای تعیین سطل یک گره جدید الورود از تعدادی نشانه استفاده می‌کنیم. گره می‌تواند آدرس IP این نشانه‌ها را از طریق DNS بدست آورد. هر گره باید این نشانه‌ها را که گره‌هایی شناخته شده و غیر از اعضاء گروه می‌باشند را Ping نماید. برای اینکه تمام گره‌های نزدیک به همدیگر در یک سطل قرار گیرند هر گره با یک ترتیب خاص و منظم این نشانه‌ها را Ping می‌نماید و تأخیر خود را نسبت به تک تک آنها محاسبه می‌نماید. سپس این اعداد را در جایگاه خود ضرب می‌نماییم به این ترتیب که عدد مربوط به اولین نشانه را در عدد یک، دومین عدد را در عدد سه، سومین عدد را در نه و چهارمین عدد را در ۲۷ ضرب می‌کنیم و نتایج آنها با همدیگر جمع می‌نماییم (اعداد در مبنای ۳ می‌باشند). درنتیجه یک عدد از انجام این محاسبات بدست می‌آید. به‌طور مثال اگر تعداد نشانه‌ها چهار باشد و تأخیر بدست آمده از هر کدام از آنها به‌ترتیب ۵۰، ۱۲۰، ۷۵ و ۲۱۰ باشد عدد بدست آمده برابر است با:

$$50\times3^0+120\times3^1+75\times3^2+210\times3^3=6577$$

بعد از بدست آوردن این اعداد آنهایی که در یک محدوده عددی مشخص قرار می‌گیرند (اگر حداکثر امتیاز را ۱۰۰۰۰ درنظر بگیریم از ۰تا ۲۵۰۰، از ۲۵۰۱ تا ۵۰۰۰ از ۵۰۰۱ تا ۷۵۰۰ و از ۷۵۰۱ تا ۱۰۰۰۰) در یک سطل قرار خواهند گرفت (شکل شماره ۱). در ابتدا ما از چهار سطل استفاده می‌نماییم، بدین ترتیب ما تضمین می‌نماییم که گره‌های نزدیک به همدیگر در یک سطل قرار خواهند گرفت. هر گره باید عدد محاسبه شده خود را ذخیره نماید.



شکل۱- نحوه کلاستر شدن گره‌ها در سطل مربوطه (اعداد نوشته شده امتیاز گره مورد نظر می‌باشند)

Archive of SID

www.SID.ir

بعد از قرار گرفتن گره‌های نزدیک به همدیگر در یک سطل، در داخل سطل از یک ساختار درختی استفاده می‌نماییم. ازآنجائی‌که دیگر تأخیر بین اعضاء موجود در سطل به‌علت نزدیک بودن این گره‌ها به همدیگر قابل چشم‌پوشی می‌باشد، گره‌ای که دارای بیشترین ماندگاری است را به‌عنوان ریشه و مسؤل انتخاب می‌نماییم.

۳–۱– اتصال گره جدید به گروه

قبل از توضیح نحوه اتصال یک گره جدید، اصطلاحات زیر را معرفی می‌نماییم.

مسؤل محلی: گره‌ای است که در سطل مربوطه به‌عنوان مسئول انتخاب می‌شود. این گره لیست تمام اعضاء موجود در سطل مربوط به خود را نگهداری می‌نماید.

مسئول میانی: گره‌ای که در مسیر بین فرستنده و مسئول محلی قرار دارد.

هنگام اتصال یک گره جدید به گروه پیغام‌های زیر بین اعضاء گروه رد و بدل می‌شود:

پیغام Build: این پیغام برای یک مسئول (محلی یا میانی) ارسال می‌گردد. گره‌ای که این پیغام را دریافت می‌نماید چک می‌کند که آیا می‌تواند گره جدید الورود را به‌عنوان فرزند خود قبول کند یا نه، اگر نتواند گره جدید را به‌عنوان فرزند خود بپذیرد شماره خود را در پیغام ثبت کرده و آن را به یکی از فرزندان خود که امتیاز گره جدیدالورود در محدوده آن می‌گنجد انتقال می‌دهد.

پیغام Build_bin: مسؤل محلی این پیغام را برای اعضاء موجود در سطل ارسال می‌نماید تا اگر دارای Fanout لازم باشند گره جدید الورود را به‌عنوان فرزند خود انتخاب نمایند.

پیغام Parent: گره ای که به‌عنوان پدر گره جدیدالورود انتخاب می‌شود این پیغام را برای آن ارسال می‌نماید.

اتصال یک گره به گروه دارای دو مرحله می‌باشد در مرحله اول با توجه به امتیاز گره مورد نظر سطل مربوطه پیدا شده، سپس در مرحله دوم در داخل سطل مورد نظر از یک ساختار درختی استفاده می‌شود. برای پیداکردن سطل مربوطه، گره مورد نظر بعد از بدست آوردن عدد (امتیاز) مربوط به خود یک پیغام Join به فرستنده[۸] ارسال می‌نماید تا آن را در سطل مناسب قرار دهد. اگر این گره اولین کسی باشد که وارد سطل می‌شود، به‌عنوان مسؤل محلی سطل مربوطه انتخاب می‌گردد. درغیراین‌صورت فرستنده با توجه به امتیاز گره مورد نظر، یکی از فرزندان خود را به آن معرفی می‌نماید و پیغام Build را برای آن ارسال می‌کند. اگر گره معرفی شده یک مسئول میانی باشد، این فرآیند تا رسیدن به یک مسئول محلی دوباره تکرار می‌گردد. یعنی یا گره یک سطل جدید را تشکیل می‌دهد یا اینکه توسط پیغام Build به یکی از سطل‌های زیرین معرفی می‌گردد. دقت کنید اگر گره مورد نظر به‌عنوان فرزند فرستنده یا یک مسؤل میانی پذیرفته شود (یعنی یک سطل جدید درست شود)، گره والد یک پیغام Parent به‌سمت گره خواهد فرستاد. دریافت پیغام Parent به‌منزله اتمام پروسه Join است. هر گره لیستی از تمام مسؤل‌های (میانی و محلی) بین فرستنده و خودش را نگهداری می‌کند تا درصورت خرابی مسئول بتوانیم با تماس با نزدیکترین مسؤل در دسترس عمل بازسازی درخت را انجام دهیم.

فرض کنید که پیغام Build به یک مسؤل محلی رسیده است. بعد از پیدا شدن مسؤل محلی و سطل مورد نظر ساختار درختی موجود در داخل سطل را می‌توان به یکی از روش‌های زیر تشکیل داد.

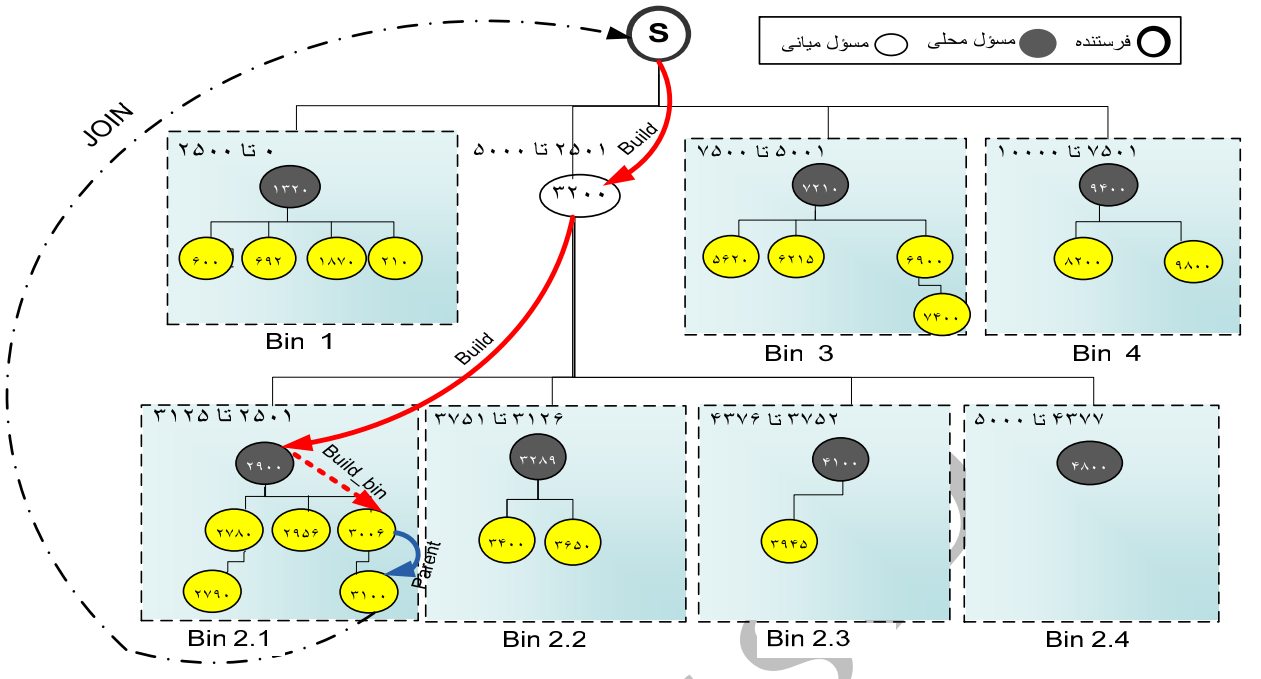
۳–۲– ساختار درخت k-ary

در داخل سطل یک ساختار درخت جستجوی k-ary (k برابر با ۴ می‌باشد) را تشکیل می‌دهیم به این ترتیب که با توجه به امتیاز گره جدیدالورود محل آن را در داخل سطل پیدا می‌نماییم. برای انجام این کار هر گره دارای یک محدوده مشخص می‌باشد و متناسب با امتیاز مربوطه $\frac{1}{4}$ از محدوده پدر خود را مورد پوشش قرار می‌دهد. درخت تشکیل شده شبیه درخت جستجوی دودویی BST است با این تفاوت که در این حالت هر گره دارای چهار فرزند می‌باشد. برای بدست آوردن محدوده مورد نظر به‌همراه پیغام Join دو متغیر با مقادیر اولیه low=0 و High=10000 (حداکثر امتیاز محدوده را ۱۰۰۰۰ درنظر گرفته‌ایم) برای فرستنده ارسال می‌شود. هنگام پیمایش درخت از ریشه به‌سمت سطل مورد نظر این متغیرها با مقادیر جدید مقداردهی می‌شوند.

به‌طور مثال در شکل شماره ۲ اگر یک گره با امتیاز ۳۱۰۰ وارد گروه شود، یک پیغام Join همراه با مقادیر Low=0 و High=10000 به فرستنده ارسال می‌نماید. فرستنده با توجه به امتیازگره جدیدالورود، گره‌ای که دارای امتیاز ۳۲۰۰ است را معرفی می‌نماید و متغیرهای Low=2501 و High=5000 را به‌همراه پیغام Build برای آن ارسال می‌نماید. این گره نیز با توجه به امتیاز مورد نظر گره ۲۹۰۰ را معرفی می‌نماید و مقادیر Low=2501 و High=3125 را به‌همراه پیغام Build برای آن ارسال می‌نماید. گره با امتیاز ۲۹۰۰ نیز گره با امتیاز ۳۰۰۶ را معرفی می‌نماید و مقادیر Low=2962 و High=3125 را به‌همراه پیغامBuild_Bin برای آن ارسال می‌نماید این گره نیز گره جدید را به لیست فرزندان خود اضافه کرده و پیغام Parent را برای آن ارسال می‌نماید. هر گره شماره خود را به پیغام ارسالی اضافه می‌نماید تا مسیر طی شده از فرستنده تا گره مورد نظر نیز ذخیره گردد. چنین ساختاری در داخل سطل باعث کاهش تنش می‌شود.

Archive of SID

www.SID.ir



شکل ۲- ساختار k-ary در داخل هر سطل

۳-۳- ساختار درخت کامل

در این روش گره جدید به‌صورت اول سطح درخت موجود در داخل سطل را پیمایش می‌نماید و اولین گره‌ای که دارای Fanout خالی است را به‌عنوان پدر خود انتخاب می‌نماید. این گره پیغام Parent را برای گره جدیدالورود ارسال و آن را به لیست فرزندان خود اضافه می‌نماید. می توان گفت که در این حالت عمل اتصال یک گره جدید به گروه دارای دو مرحله متفاوت می‌باشد. در مرحله اول با توجه به امتیاز گره مورد نظر، سطل مربوطه انتخاب می‌شود و در مرحله دوم بدون توجه به امتیاز گره مورد نظر در داخل سطل عمل جستجوی اول سطح انجام می‌شود تا یک گره با Fanout لازم پیدا شود. این روش باعث کاهش مؤثر عمق درخت و درنتیجه کاهش تأخیر می‌شود.

درخصوص درخت ساخته شده ذکر نکات زیر ضروری به‌نظر می‌رسد:

در ساخت درخت سعی می‌شود که درخت حاصل دارای عمق $\log_m^N$ باشد. که N تعداد گیرنده‌ها و m تعداد سطل‌ها است.

عمل Join با تعداد ثابتی Ping انجام می‌پذیرد، و سربار زیادی برروی گروه ایجاد نمی‌کند.

هر گره باید لیست فرزندان، مسؤل محلی و مسؤل‌های میانی مسیر خود را درصورت وجود نگهداری کند.

هر مسؤل محلی لیست اطلاعات تمام گره‌های موجود در سطل مربوطه به‌همراه امتیاز و زمان ورود آنها را نگهداری می‌نماید.

استفاده از درخت k-ary در داخل سطل باعث کاهش تنش و استفاده از درخت کامل باعث کاهش تأخیر می‌شود.

۳-۴- انجام عمل شکاف

در حالت شروع ما از تعداد چهار سطل برای ساخت درخت استفاده می‌نماییم. در صورتی که تعداد گره‌های موجود در یک سطل بیشتر از یک حد مشخص گردد، آن سطل را به تعداد بیشتری می‌شکنیم (عمل شکاف). در هر بار شکستن سطل، چهار سطل جدید ایجاد می‌گردد. مزایای این کار عبارتند از:

عمق درخت ساخته شده خیلی زیاد نخواهد شد.

سطل‌های به‌وجود آمده تا مدت زمان زیادی پر نخواهند شد و عمل شکاف بعدی دیرتر صورت می‌پذیرد.

برای انجام شکاف در یک سطل، ابتدا مسؤل محلی پیغام Split را به تمام گره‌های موجود در سطل ارسال می‌نماید. گره‌های موجود بعد از دریافت پیغام Split یک پیغام Join به‌سمت مسؤل محلی (نه فرستنده) خود می‌فرستند و عدد بدست آمده از مرحله Join را به آن اعلام می‌نمایند. مسؤل محلی با استفاده از این اعداد، گیرنده‌ها را درسطل جدید از زیر مجموعه خود قرار می‌دهد و خود به یک مسئول میانی تبدیل می‌شود. بعد از قرارگرفتن گره‌ها در سطل‌های جدید، عمل انتخاب مسئول محلی جدید برای هرکدام از این سطل-ها صورت می‌گیرد. این مسئول‌های محلی جدید به مسئول میانی (مسئول محلی قبلی) متصل می‌شوند. با این عمل محدوده قبلی

Archive of SID
www.SID.ir

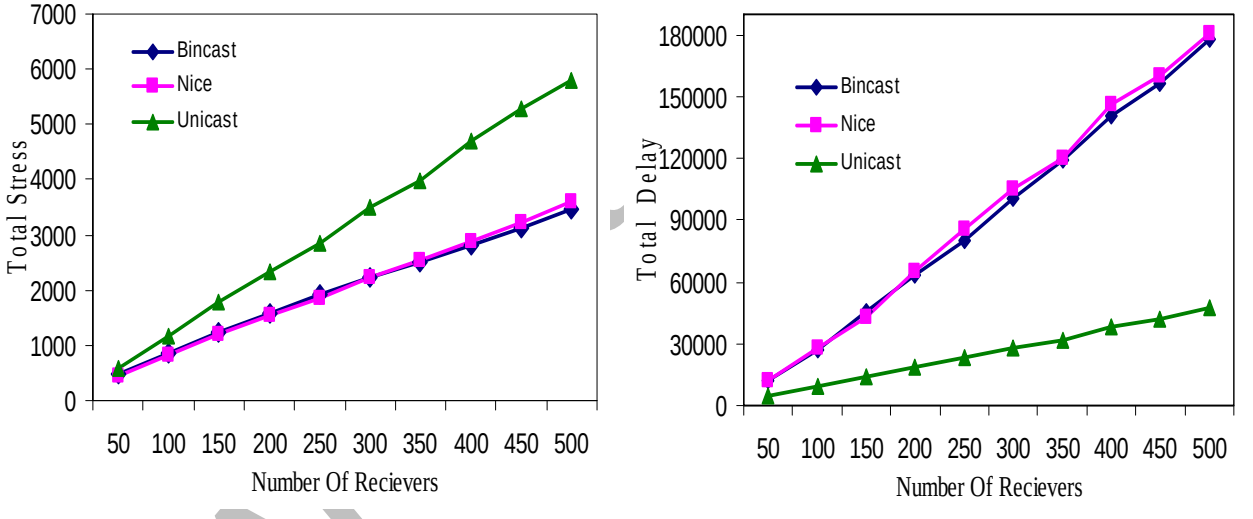
عملاً به چهارقسمت دیگر تقسیم می‌شود. شکل شماره ۲ عمل شکاف در سطل دوم (محدوده 2501 تا 5000) را نشان می‌دهد. در این حالت چهار سطل جدید با محدوده از ۲۵۰۱ تا ۳۱۲۵، از ۳۱۲۶ تا ۳۷۵۱، از ۳۷۵۲ تا ۴۳۷۶و از ۴۳۷۷ تا ۵۰۰۰ تشکیل می‌شود.

۳–۵– خروج (خرابی) یک میزبان

هر گره با والد و فرزندان خود در سطل به‌صورت دوره پیغام زنده بودن (Hello) را رد و بدل می‌کند. اگر گره‌ای بخواهد سطل مربوطه را ترک کند پیغام Remove را به گره‌های همسایه خود ارسال می‌نماید. بعد از ترک یک سطل توسط گره غیربرگ، باید کلیه فرزندان آن گره دوباره عمل Join را با اولین مسئول در دسترس انجام دهند (لیست مسئول‌ها هنگام Join توسط گره ذخیره می‌شود). از آنجایی‌که دیگر نیازی به Ping کردن نشانه‌ها نیست، می‌توان گفت که سرعت انجام این Join خیلی بیشتر از Join اولیه می‌باشد. اگر یک گره خراب شود و سطل مربوطه را بدون ارسال پیغام Remove ترک نماید، گره‌های همسایه آن با توجه به عدم دریافت چندین پیغام Hello متوالی، متوجه موضوع خواهند شد و عمل ترمیم درخت را شروع می‌نمایند.

۴– نتایج شبیه‌سازی

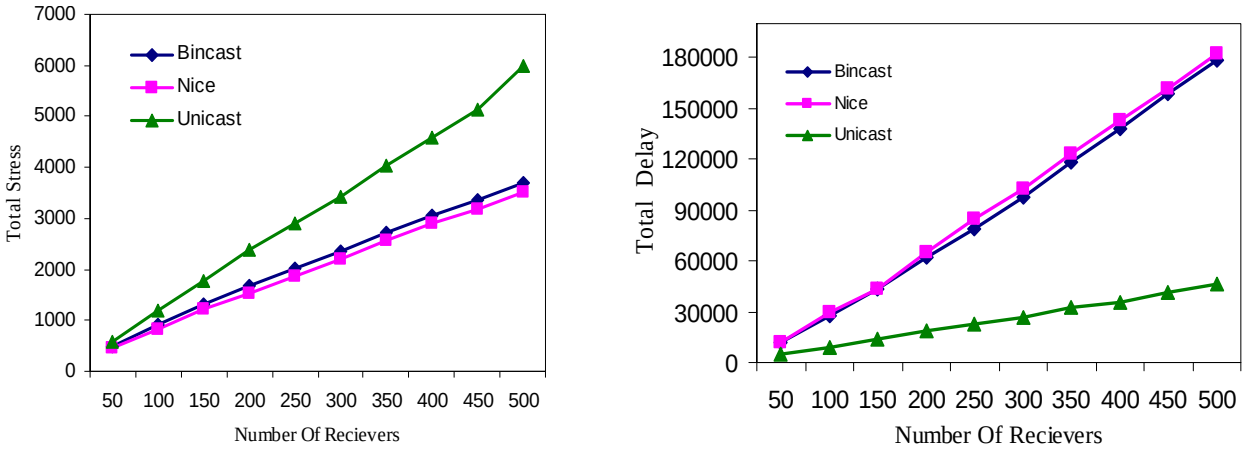
در این بخش نتایج شبیه‌سازی ارائه می‌شود. از تولیدکننده توپولوژی GT_ITM و مدل Transit_Stub برای تولید گراف‌های تصادفی استفاده شده است [۶]. هرگراف دارای ۱۰۱۰۰ روتر با Fanout بین ۳ و ۴ می‌باشد. تمام میزبان‌ها به‌صورت تصادفی از بین نودهای موجود در نواحی Stub انتخاب می‌شوند. ما تعداد سطل‌های مورد نظر را چهار و هشت انتخاب کرده‌ایم. در داخل هر سطل حداکثر ۱۰ گره موجود می‌باشد. الگوریتم مورد نظر با NICE که یکی از بهترین روش‌های موجود می‌باشد مقایسه شده است. در نمودارهای ارائه شده، هر نقطه متوسط ۱۰۰ بار اجرای شبیه‌ساز می‌باشد. برای انجام اینکار از ده شبیه‌سازی تصادفی و متفاوت در ده گراف مختلف استفاده شده است.



شکل ۳ – درخت k-ary با چهار سطل الف) مجموع تأخیر تمام گیرنده‌ها ب) مجموع تنش تمام گیرنده‌ها

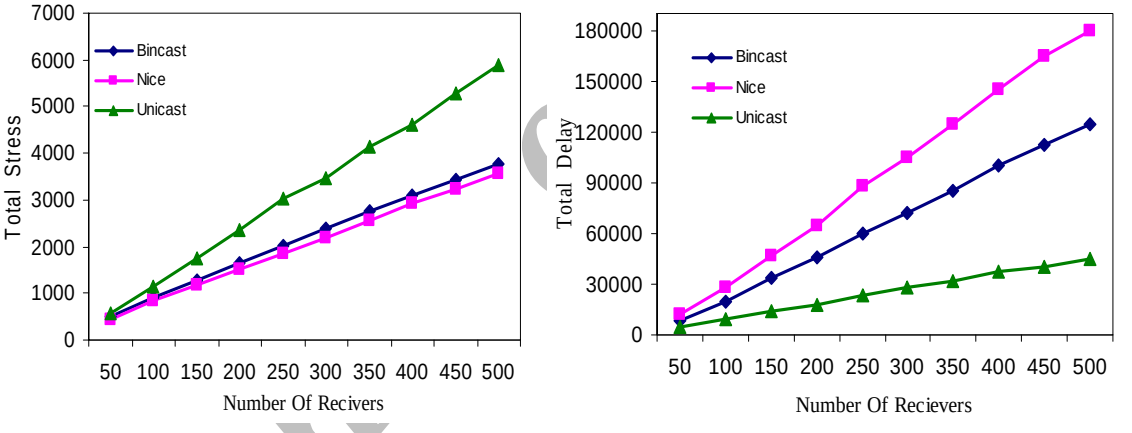
قسمت‌های الف و ب شکل سه به‌ترتیب مجموع تأخیر و تنش را برای تمام اعضا گروه نشان می‌دهد. در این شکل تعداد گیرنده‌ها بین ۵۰ تا ۵۰۰ درنظر گرفته شده است. هر سطح الگوریتم Bincast دارای چهار سطل و در داخل هر سطل از ساختار k-ary استفاده شده است. همان‌طور که مشاهده می‌گردد Bincast و Nice هم در مورد تأخیر و هم در مورد تنش بسیار شبیه به هم عمل می‌نمایند. البته با افزایش سایز گروه، Bincast کمی بهتر از Nice عمل می‌کند. در قسمت‌های الف و ب شکل چهار به‌ترتیب تأخیر و تنش کل را برای حالتی که چهار سطل وجود دارد و در داخل هر سطل از ساختار درخت کامل استفاده شده است نشان می‌دهند. مشاهده می‌شود که رفتار Nice و Bincast باز شبیه هم است با این تفاوت که تنش Nice کمی بهتر است. البته در مقابل تأخیر Bincast تا حدودی بهتر است.

www.SID.ir



شکل ۴- درخت کامل با چهار سطل الف) مجموع تأخیر تمام گیرنده‌ها ب) مجموع تنش تمام گیرنده‌ها

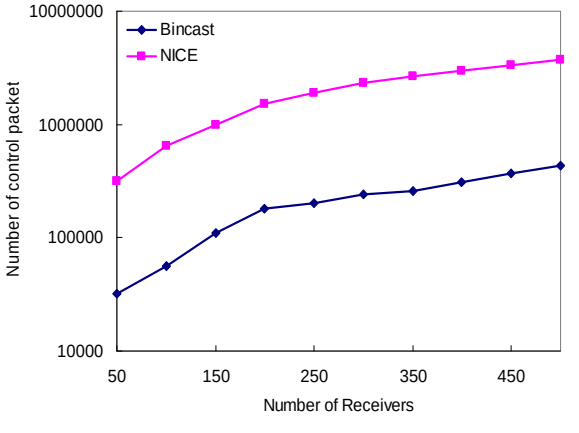
قسمت‌های الف و ب شکل (۵) به‌ترتیب تأخیر و تنش کل را برای حالتی که هشت سطل وجود دارد و در داخل هر سطل از ساختار درخت k-ary استفاده شده است نشان می‌دهند.



شکل ۵- درخت k-ary با هشت سطل الف) مجموع تأخیر تمام گیرنده‌ها ب) مجموع تنش تمام گیرنده‌ها

همان‌طور که مشاهده می‌شود استفاده از هشت سطل باعث کاهش مؤثر تأخیر کل گردیده است بدون اینکه تنش تغییر قابل ملاحظه‌ای داشته باشد. یعنی می‌توان با تأخیر خیلی کمتر نیز گروه چندپخشی را ایجاد نمود. کاهش تأخیر به‌علت کاهش عمق درخت از $\log_4^N$ به $\log_8^N$ می‌باشد. افزایش بیشتر تعداد سطل‌ها اگر چه باز هم باعث کاهش تأخیر می‌شود، اما تنش را نیز به‌طور قابل ملاحظه‌ای افزایش خواهد داد. نمودارهای مربوط به درخت کامل هشت سطلی نیز شبیه شکل پنج هستند و رفتار مشابهی را نشان می‌دهند. با این تفاوت که تنش Bincast از Nice کمی بهتر است.

در شکل (۶) مقیاس‌پذیری دو روش Nice و Bincast مورد مطالعه قرار می‌گیرد. برای این منظور، تعداد پیغام‌های ارسال شده بین گره‌ها برای ساخت و نگهداری درخت در مقیاس لگاریتمی آورده شده است. همان‌طور که مشاهده می‌کنید تعداد پیغام‌های ارسال شده در Bincast بسیار کمتر می‌باشد و این نشان‌دهنده مقیاس‌پذیری زیاد و سربار کم روش پیشنهادی می‌باشد.



شکل ۶– سربار کنترلی روش پیشنهادی

www.SID.ir

۵– نتیجه‌گیری

روش‌های سلسله مراتبی و براساس درخت به‌خاطر سربار کنترلی کم و کارایی مناسب بسیار مقیاس‌پذیر می‌باشند [۱۷]. در این مقاله ما یک روش جدید در حیطه ALM ارائه نمودیم که گره‌های نزدیک به همدیگر را با روش سطل‌بندی در یک سطل قرار می‌دهد و دارای سربار کنترلی کمی نسبت‌به روش‌های مشابه می‌باشد. در این روش هر گره با تعداد ثابتی Ping می‌تواند موقعیت خود را پیدا کرده و به گروه متصل شود. از طرفی لیست تعداد کمی از دیگر گره‌ها را نگهداری می‌نماید. تمرکز اصلی ما بیشتر بر روی تحمل‌پذیری خطا و تأخیر کمتر بود. در Bincast گره‌هایی که به‌عنوان مسئول میانی یا محلی انتخاب می‌گردند دارای پایداری بیشتری هستند. درحالی‌که در روش‌های مشابه گره‌هایی که دارای fanout بیشتر یا نزدیک‌تر به منبع هستند را در سطوح بالاتر درخت قرار می‌دهند. این کار اگر چه ممکن است کیفیت درخت را بهبود دهد اما پایداری درخت را در برابر خرابی گره کاهش می‌دهد. در مقابل، Bincast با حفظ معیار پایداری به تأخیر کمتر و تنش مشابه دست یافته است. ضمن این‌که Bincast با انتخاب پایدارترین گره به‌عنوان مسئول محلی و نگهداری لیست مربوط به مسئول‌های موجود در مسیر گره تا فرستنده دارای تحمل‌پذیری زیادی در برابر خطا می‌باشد.

۶– منابع

[1] Y. H. Chu, S. G. Rao, H. Zhang; **"A Case for End System Multicast"**, In Proceeding of ACM SIGMETRICS, June 2000.

[2] S .Banerjee, B. Bhattachariee, C.Kommaredy; **"Scalable Application Layer Multicast"**, In Proceeding of ACM Sigcomm, August 2002.

[3] B. Zhang, S. Jamin, L. Zhang; **"Host Multicast:A Framwork for Delivering Multicast to End User"**, Proc of IEEE INFOCOM'02, June 2002.

[4] P. Francis, **"Yoid:_Extending the Internet Multicast Architecture"**, ACIRI April,2002.

[5] S. Savage, A. Collins, E. Ho_man, J. Snell, T. Anderson; **"The End-to-End Effects of Internet Path Selection"**, SIGCOMM Computer Communication Review, Vol. 29, No. 4, pp. 289–299, 1999.

[6] M. Castro, P. Druschel, A-m. Kermarree, A. Rowstron, **"SCRIBE: A Large Scale and Decentralized Application Level Multicast Infrastructure"**, IEEE Journal on Selected Areas in communications (JSAC), Vol. 20, No.8. 2002, pp. 1489-1499.

[7] M. kwon, S. Fahmy; **"Topology-Aware Overlay Network for Group Communication"**, Miami Florida USA, NOSSDAV'02 May 12-13, 2002.

[8] C. Kommareddy, N. Shankar, B. Bhattacharjee, **"Finding Close Friends on the Internet"**, ICNP, November 2001.

[9] S. Horng, L. Chun, C. Yang, H-L Hsu, **"A DNS-aided Application Layer Multicast Protocol"**, LAENG international Conference on Communication System and Application IMECS , Hong Kong, Vol. 2, March, 2008 pp. 1076-1082.

[10] B. Y. Zhao, J. Kubiatowicz, A. D. Joseph; **"Tapestry:An Infrastructure for Fault-Tolerant Wide-area Location and Routing"**, Tech. Rep. UCB/CSD-01-1141, University of California at Berkeley, Computer Science Division, April 2001.

[11] A. Rowstron, P. Druschel; **"Pastry:Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems"**, In IFIP ACM International Conference on Distributed System Platform (Middleware), November 2001.

[12] S. Ratnasamy, M. Handly, R. Karp, S. Shenker; **"Topologically-Aware Overlay Construction and Server Selection"**, In Proceedings of INFOCOM, June 2002.

[13] B. Y. Zhao, Y. Duan, L. Huang, A. Joseph, and J. Kubiatowicz; **"Brocade: Landmark Routing on Overlay Networks"**, in 1st International Workshop on Peer-to-Peer Systems (IPTPS'02), 2002

[14] Z. Xu, M. Mahalingam, M. Karlsson; **"Turning Heterogeneity into an Advantage in Overlay Routing"**, IEEE INFOCOM, Vol. 2, pp. 1499-1509, San Francisco, CA, 2003.

[15] C. Wu1, D. Liu and R. Hwang; **"A Location-Aware Peer-to-Peer Overlay Network"**, Int. J. Communication Systems, No. 20, pp.83–102, 2007;

[16] K. Yusung, C. Kilnam; **"Scalable and Topologically-Aware Application Layer Multicast"**, IEEE GLOBECOM, Vol. 2, pp.1266 – 1270, 2004.

[17] S. Banerjee, B. Bhattacharjee; **"Comparative Study of Application Multicast Protocols"**, Available from:http://www.cs.umd.edu /users/suman/pubs/compare.ps

[18] K. Calvert, E. Zegura, S. Bhattacharjee; **How to Model an Internetwork**, In Proceeding of IEEE Infocom, 1996.

[19] S. Banerjee, myns Simulator, Available from: http://www.cs.umd.edu/~suman/research/myns/index.html.

Archive of SID

www.SID.ir

۷– پی‌نوشت‌ها

1- Multicast
2- Stress
3- Binning
4- Landmark
5- Split
6- locality
7- Beaconing
8- Source

Archive of SID

www.SID.ir