

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، در سال‌های اخیر روش‌های جایگزین زیادی در حوزه Application Layer multicast (ALM) به وجود 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. از طرفی مجاورت گره‌ها باعث می‌شود زیردرخت مذکور دارای تنش زیادی نباشد. لذا درخت نهایی دارای تأخیر و تنش کمی است. از آنجایی که گره‌های ماندگارتر به‌عنوان مسؤل در نظر 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

مربوط به گروه و پیاده‌سازی چند پخشی در مسیریاب‌ها غیرممکن به نظر 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

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

این روش‌ها را بررسی می‌کنیم. در HMTTP [۳] از یک ساختار درختی برای اتصال میزبان‌های شرکت کننده در گروه استفاده می‌شود. در این روش با آنکه تنش در لینک‌های موجود حداقل می‌باشد اما از لحاظ تحمل خطا (ترک یک یا چند گره) دارای تحمل پذیری 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 شرکت کننده 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 چند گره) دارای تحمل بیشتری می‌باشد، اما از لحاظ مقیاس پذیری 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 در نظر گرفته می‌شود و گره وسط کلاستر (از لحاظ فاصله) انتخاب می‌گردد. در سطوح بالاتر مسؤل‌های کلاسترها به صورت 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 نظیر به نظیر (p2p) همانند Pastry [۱۱] و Tapestry [۱۰] برای ساخت overlay و مسیریابی، به صورت 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 نظیر به نظیر t-CAN [۱۲] و eCAN [۱۴] به صورت صریح از اطلاعات توپولوژی استفاده می‌کنند. CAN - فضای مورد نظر را همانند یک فضای کارترین در نظر می‌گیرد و به هر گره یک محدوده از این فضا را نسبت می‌دهد و از ترتیب نشانه‌ها برای کلاستر کردن گره‌ها موجود در شبکه استفاده می‌کند. eCAN با استمداد از

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

در روش پیشنهادهی (Bincast)، با استفاده از تکنیک سطل بندی تمام گره‌های نزدیک به همدیگر با کمک تعداد ثابتی نشانه^۴ که گره‌های شناخته شده‌ای می‌باشند در یک سطل قرار می‌گیرند. بعد از قرار گرفتن 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 مسؤل به عنوان ریشه زیردرخت انتخاب می‌گردد. با توجه به مجاورت گره‌های هر سطل تأخیر زیر درخت ساخته شده کمتر از روش‌های دیگر می‌باشد. از طرفی مجاورت گره‌ها باعث می‌شود که زیردرخت مذکور دارای تنش زیادی نباشد. لذا درخت نهایی دارای تأخیر و تنش کمی است. از آنجایی که گره‌های ماندگارتر به عنوان مسؤل در نظر گرفته می‌شوند می‌توان گفت که این روش نسبت به روش‌های دیگر دارای تحمل پذیری بیشتری در برابر خطا (ترک گره) می‌باشد. نهایتاً با توجه به این که از روش سطل بندی استفاده می‌نماییم و در هر سطل تعداد زیادی گره می‌تواند قرار گیرد مقیاس پذیری روش ارائه شده زیاد می‌باشد.

در ادامه در بخش دوم کارهای مرتبط عنوان شده است. در بخش سوم روش پیشنهادی به صورت کامل شرح داده شده است و نحوه اتصال گره جدید به گروه و انجام عمل شکاف^۵ مطرح می‌شود. در بخش چهارم نتایج شبیه سازی و در بخش پنجم نتیجه گیری 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

می تواند آدرس IP این نشانه ها را از طریق DNS بدست آورد. هر گره باید این نشانه ها را که گره هایی شناخته شده و غیر از اعضا گروه می باشند را Ping نماید. برای اینکه تمام گره های نزدیک به همدیگر در یک سطل قرار گیرند هر گره با یک ترتیب خاص و منظم این نشانه ها را Ping می نماید و تأخیر خود را نسبت به تک تک آنها محاسبه می نماید. سپس این اعداد را در جایگاه خود ضرب می نماییم 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 می کنیم و نتایج آنها با همدیگر جمع می نماییم (اعداد در مبنای ۳ می باشند). در نتیجه یک عدد از انجام این محاسبات بدست می آید. به طور مثال اگر تعداد نشانه ها چهار باشد و تأخیر بدست آمده از هر کدام از آنها به ترتیب ۵۰، ۱۲۰، ۷۵ و ۲۱۰ باشد عدد بدست آمده برابر است با:

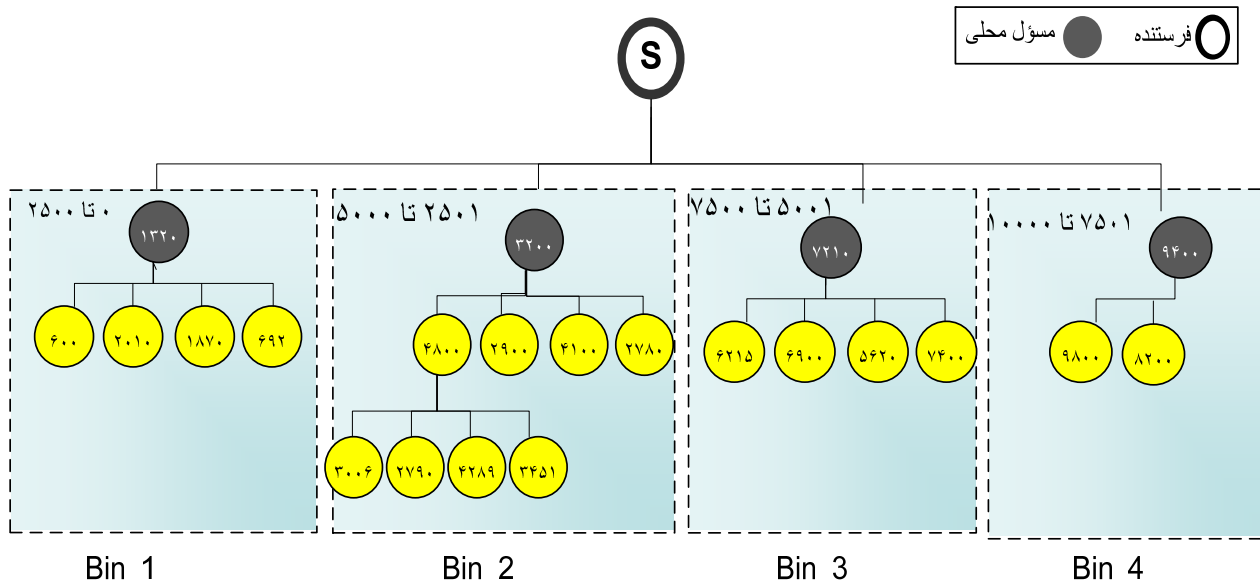
$$50 \times 3^0 + 120 \times 3^1 + 75 \times 3^2 + 210 \times 3^3 = 6577$$

بعد از بدست آوردن این اعداد آنهایی که در یک محدوده عددی مشخص قرار می گیرند (اگر حداکثر امتیاز را ۱۰۰۰۰ در نظر بگیریم از ۰ تا 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 می نماییم، بدین ترتیب ما تضمین می نماییم که گره های نزدیک به همدیگر در یک سطل قرار خواهند گرفت. هر گره باید عدد محاسبه شده خود را ذخیره نماید.

نشانه ها، یک شبکه overlay تحت عنوان expressway، برای بهبود مسیریابی در CAN می سازد. Brocade [۱۳] یک شبکه overlay بر روی Tapestry می سازد و از خاصیت "محلّی بودن" مسیریابی در سطح AS استفاده می کند. [۷] Tag از هم پوشانی مسیر فرستنده در گیرنده های مختلف برای ساخت overlay استفاده می کند. Laptop [۱۵] از مکان جغرافیای نودها استفاده کرده و یک شبکه سلسله مراتبی را می سازد. در این الگوریتم هر گره موقعیت خود را با Ping کردن تعداد کوچکی از گره های موجود در شبکه بدست می آورد. الگوریتم مطرح شده در [۸] از مفهوم نشانه^۷ برای کلاستر کردن گره های موجود در شبکه استفاده می کند در حالی که الگوریتم موجود در [۹] از DNS برای کلاستر کردن استفاده می کند و گره های را که از لحاظ DNS به یکدیگر نزدیک باشند در یک کلاستر قرار می دهد. در [۱۶] از یک سری نشانه برای کلاستر کردن استفاده می شود در این روش هر گره تعدادی نشانه را Ping می کند و براساس ترتیب دریافت جواب از آنها کار کلاستر کردن را انجام می دهد و گره های که دارای یک ترتیب یکسان باشند را در یک کلاستر قرار می دهد.

۳- روش پیشنهادی

ابتدا مکانیسم سطل بندی را توضیح می دهیم. برای تعیین سطل یک گره جدید ورود از تعدادی نشانه استفاده می کنیم. گره



شکل ۱- نحوه کلاستر شدن گره ها در سطل مربوطه (اعداد نوشته شده امتیاز گره مورد نظر می باشند)

معرفی می گردد. دقت کنید اگر گره مورد نظر به عنوان فرزند فرستنده یا یک مسؤل میانی پذیرفته شود (یعنی یک سطل جدید درست شود)، گره والد یک پیغام Parent به سمت گره خواهد فرستاد. دریافت پیغام Parent به منزله اتمام پروسه Join است. هر گره لیستی از تمام مسؤل های (میانی و محلی) بین فرستنده و خودش را نگهداری می کند تا در صورت خرابی مسؤل 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 می توان به یکی از روش های زیر تشکیل داد.

۳-۲- ساختار درخت k-ary

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

به طور 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 می نماید. فرستنده با توجه به امتیاز گره جدیدالورود، گره ای که دارای امتیاز ۳۲۰۰ است را معرفی می نماید و متغیرهای Low=2501 و High=5000 را به همراه پیغام Build برای آن ارسال می نماید. این گره نیز با توجه به امتیاز مورد نظر گره ۲۹۰۰ را معرفی می نماید و مقادیر Low=2501 و High=3125 را به همراه پیغام Build برای آن ارسال می نماید. گره با امتیاز ۲۹۰۰ نیز گره با امتیاز ۳۰۰۶ را معرفی می نماید و مقادیر Low=2962 و High=3125 را به همراه پیغام Build_Bin برای آن ارسال می نماید این گره نیز گره جدید را به لیست فرزندان خود اضافه کرده و پیغام Parent را برای آن ارسال می نماید. هر گره شماره خود را به پیغام ارسالی اضافه می نماید تا مسیر طی شده از فرستنده تا گره مورد نظر نیز ذخیره گردد. چنین ساختاری در داخل سطل باعث کاهش تنش می شود.

بعد از قرار گرفتن گره های نزدیک به همدیگر در یک سطل، در داخل سطل از یک ساختار درختی استفاده می نماییم. از آنجائی که دیگر تأخیر بین اعضاء موجود در سطل به علت نزدیک بودن این گره ها به همدیگر قابل چشم پوشی می باشد، گره ای که دارای بیشترین ماندگاری است را به عنوان ریشه و مسؤل انتخاب می نماییم.

۳-۱- اتصال گره جدید به گروه

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

مسؤل محلی: گره ای است که در سطل مربوطه به عنوان مسؤل انتخاب می شود. این گره لیست تمام اعضاء موجود در سطل مربوط به خود را نگهداری می نماید.

مسؤل میانی: گره ای که در مسیر بین فرستنده و مسؤل محلی قرار دارد.

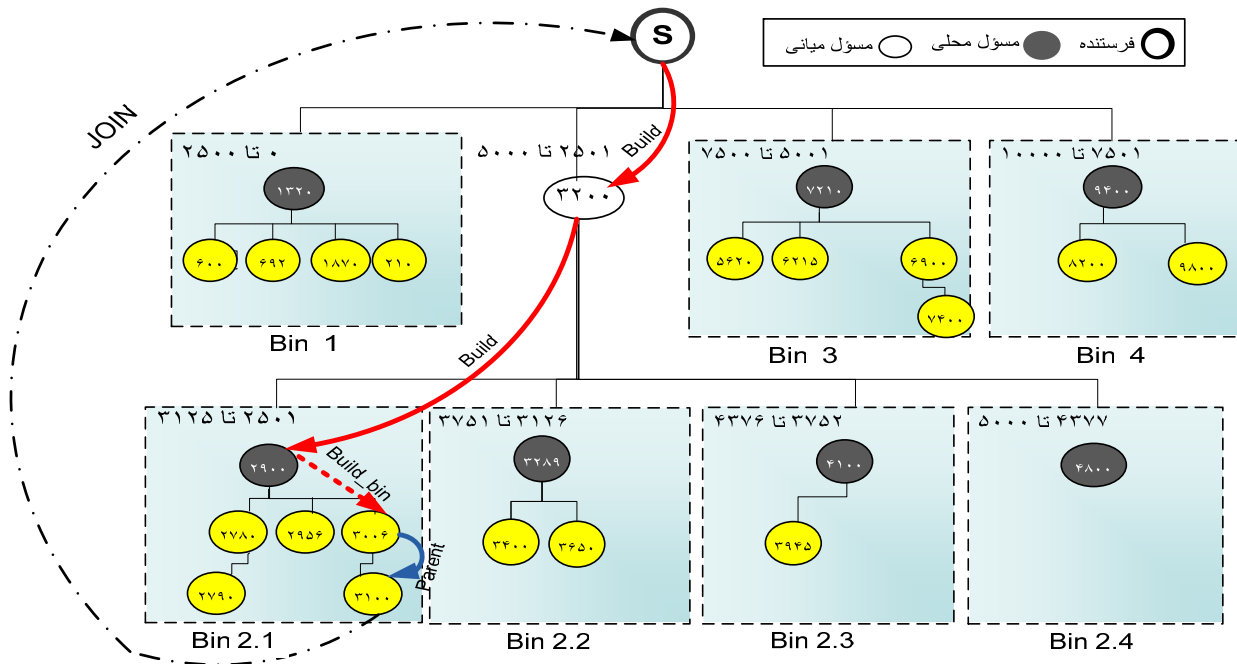
هنگام اتصال یک گره جدید به گروه پیغام های زیر بین اعضاء گروه رد و بدل می شود:

پیغام Build: این پیغام برای یک مسؤل (محلی یا میانی) ارسال می گردد. گره ای که این پیغام را دریافت می نماید چک می کند که آیا می تواند گره جدیدالورود را به عنوان فرزند خود قبول کند یا نه، اگر نتواند گره جدید را به عنوان 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 می گنجد انتقال می دهد.

پیغام Build_bin: مسؤل محلی این پیغام را برای اعضاء موجود در سطل ارسال می نماید تا اگر دارای Fanout لازم باشند گره جدیدالورود را به عنوان فرزند خود انتخاب نمایند.

پیغام Parent: گره ای که به عنوان پدر گره جدیدالورود انتخاب می شود این پیغام را برای آن ارسال می نماید.

اتصال یک گره به گروه دارای دو مرحله می باشد 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 می شود. برای پیدا کردن سطل مربوطه، گره مورد نظر بعد از بدست آوردن عدد (امتیاز) مربوط به خود یک پیغام Join به فرستنده^۸ ارسال می نماید تا آن را در سطل مناسب قرار دهد. اگر این گره اولین کسی باشد که وارد سطل می شود، به عنوان مسؤل محلی سطل مربوطه انتخاب می گردد. در غیر این صورت فرستنده با توجه به امتیاز گره مورد نظر، یکی از فرزندان خود را به آن معرفی می نماید و پیغام Build را برای آن ارسال می کند. اگر گره معرفی شده یک مسؤل میانی باشد، این فرآیند تا رسیدن به یک مسؤل محلی دوباره تکرار می گردد. یعنی یا گره یک سطل جدید را تشکیل می دهد یا اینکه توسط پیغام Build به یکی از سطل های زیرین



شکل ۲- ساختار k-ary در داخل هر سطل

مربوطه به همراه امتیاز و زمان ورود آنها را نگهداری می نماید. استفاده از درخت k-ary در داخل سطل باعث کاهش تنش و استفاده از درخت کامل باعث کاهش تأخیر می شود.

۴-۴- انجام عمل شکاف

در حالت شروع ما از تعداد چهار سطل برای ساخت درخت استفاده می نماییم. در صورتی که تعداد گره های موجود در یک سطل بیشتر از یک حد مشخص گردد، آن سطل را به تعداد بیشتری می شکنیم (عمل شکاف). در هر بار شکستن سطل، چهار سطل جدید ایجاد می گردد. مزایای این کار عبارتند از: عمق درخت ساخته شده خیلی زیاد نخواهد شد. سطل های به وجود آمده تا مدت زمان زیادی پر نخواهند شد و عمل شکاف بعدی دیرتر صورت می پذیرد.

برای انجام شکاف در یک سطل، ابتدا مسؤل محلی پیغام Split را به تمام گره های موجود در سطل ارسال می نماید. گره های موجود بعد از دریافت پیغام Split یک پیغام Join به سمت مسؤل محلی (نه فرستنده) خود می فرستند و عدد بدست آمده از مرحله Join را به آن اعلام می نمایند. مسؤل محلی با استفاده از این اعداد، گیرنده ها را در سطل جدید از زیر مجموعه خود قرار می دهد و خود به یک مسؤل میانی تبدیل می شود. بعد از قرار گرفتن گره ها در سطل های جدید، عمل انتخاب مسؤل محلی جدید برای هر کدام از این سطل ها صورت می گیرد. این مسؤل های محلی جدید به مسؤل میانی (مسؤل محلی قبلی) متصل می شوند. با این عمل محدوده قبلی

۳-۳- ساختار درخت کامل

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

در خصوص درخت ساخته شده ذکر نکات زیر ضروری به نظر می رسد:

در ساخت درخت سعی می شود که درخت حاصل دارای عمق \log_m^N باشد. که N تعداد گیرنده ها و m تعداد سطل ها است. عمل Join با تعداد ثابتی Ping انجام می پذیرد، و سربار زیادی بر روی گروه ایجاد نمی کند.

هر گره باید لیست فرزندان، مسؤل محلی و مسؤل های میانی مسیر خود را در صورت وجود نگهداری کند.

هر مسؤل محلی لیست اطلاعات تمام گره های موجود در سطل

عدم دریافت چندین پیغام Hello متوالی، متوجه موضوع خواهند شد و عمل ترمیم درخت را شروع می نمایند.

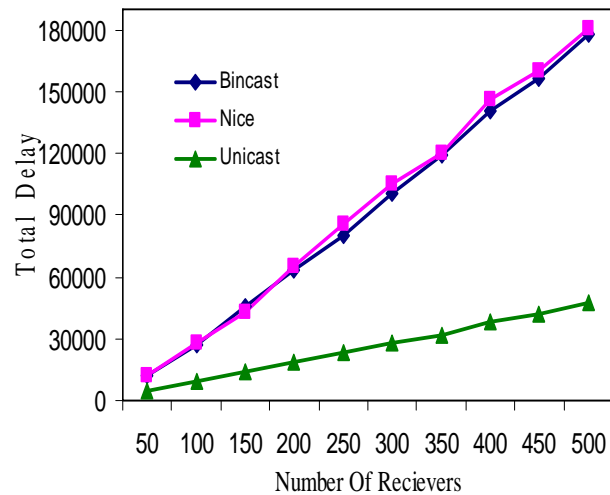
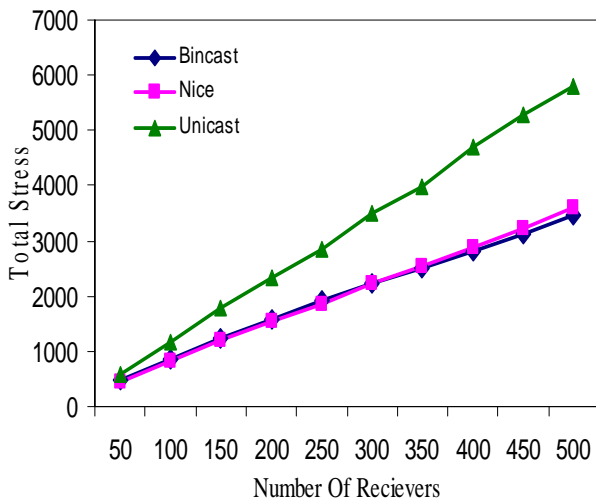
۴- نتایج شبیه سازی

در این بخش نتایج شبیه سازی ارائه می شود. از تولیدکننده توپولوژی GT_ITM و مدل Transit_Stub برای تولید گراف های تصادفی استفاده شده است [۶]. هر گراف دارای ۱۰۱۰۰ روتر با Fanout بین ۳ و ۴ می باشد. تمام میزبان ها به صورت تصادفی از بین نودهای موجود در نواحی Stub انتخاب می شوند. ما تعداد سطل های مورد نظر را چهار و هشت انتخاب کرده ایم. در داخل هر سطل حداکثر ۱۰ گره موجود می باشد. الگوریتم مورد نظر با NICE که یکی از بهترین روش های موجود می باشد مقایسه شده است. در نمودارهای ارائه شده، هر نقطه متوسط ۱۰۰ بار اجرای شبیه ساز می باشد. برای انجام اینکار از ده شبیه سازی تصادفی و متفاوت در ده گراف مختلف استفاده شده است.

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

۳-۵- خروج (خرابی) یک میزبان

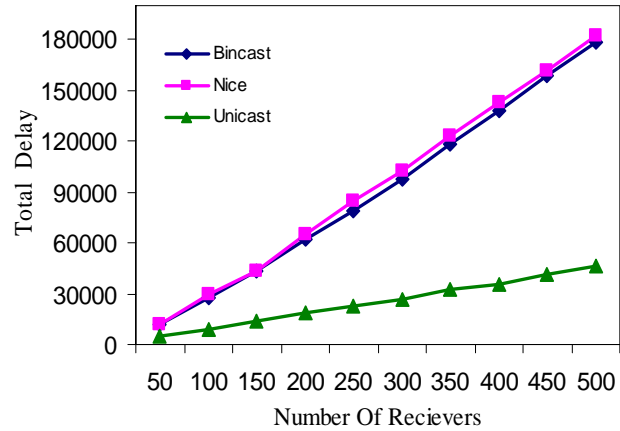
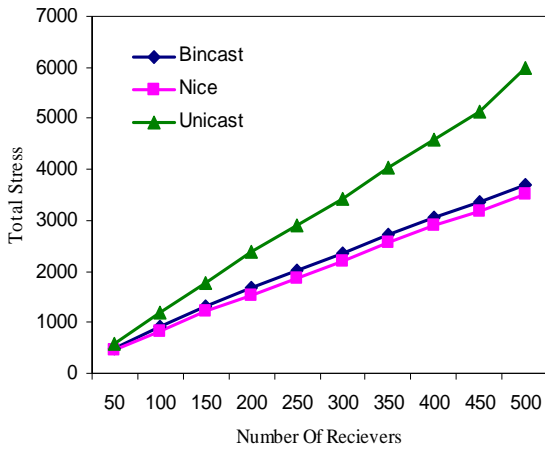
هر گره با والد و فرزندان خود در سطل به صورت دوره پیغام زنده بودن (Hello) را رد و بدل می کند. اگر گره ای بخواهد سطل مربوطه را ترک کند پیغام Remove را به گره های همسایه خود ارسال می نماید. بعد از ترک یک سطل توسط گره غیربرگ، باید کلیه فرزندان آن گره دوباره عمل Join را با اولین مسئول در دسترس انجام دهند (لیست مسئول ها هنگام Join توسط گره ذخیره می شود). از آنجایی که دیگر نیازی به Ping کردن نشانه ها نیست، می توان گفت که سرعت انجام این Join خیلی بیشتر از Join اولیه می باشد. اگر یک گره خراب شود و سطل مربوطه را بدون ارسال پیغام Remove ترک نماید، گره های همسایه آن با توجه به



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

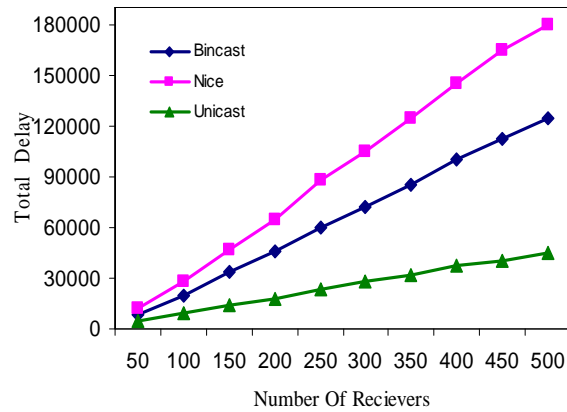
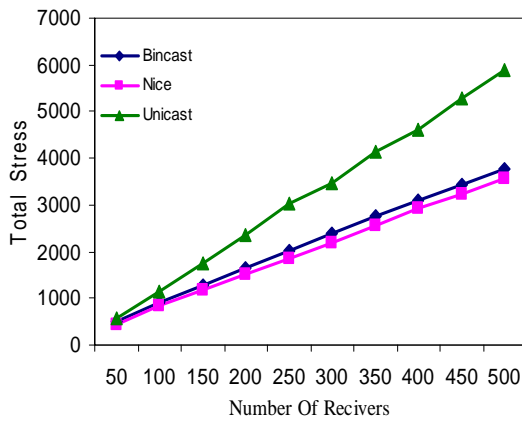
Nice عمل می کند. در قسمت های الف و ب شکل چهار به ترتیب تأخیر و تنش کل را برای حالتی که چهار سطل وجود دارد و در داخل هر سطل از ساختار درخت کامل استفاده شده است نشان می دهند. مشاهده می شود که رفتار Nice و Bincast باز شبیه هم است با این تفاوت که تنش Nice کمی بهتر است. البته در مقابل تأخیر Bincast تا حدودی بهتر است.

قسمت های الف و ب شکل سه به ترتیب مجموع تأخیر و تنش را برای تمام اعضا گروه نشان می دهد. در این شکل تعداد گیرنده ها بین ۵۰ تا ۵۰۰ در نظر گرفته شده است. هر سطح الگوریتم Bincast دارای چهار سطل و در داخل هر سطل از ساختار k-ary استفاده شده است. همان طور که مشاهده می گردد Bincast و Nice هم در مورد تأخیر و هم در مورد تنش بسیار شبیه به هم عمل می نمایند. البته با افزایش سایز گروه، Bincast کمی بهتر از



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

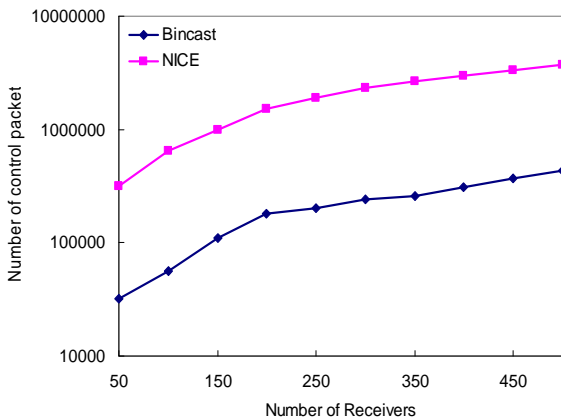
قسمت های الف و ب شکل (۵) به ترتیب تأخیر و تنش کل را برای حالتی که هشت سطل وجود دارد و در داخل هر سطل از ساختار درخت k-ary استفاده شده است نشان می دهند.



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

ارسال شده در Bincast بسیار کمتر می باشد و این نشان دهنده مقیاس پذیری زیاد و سربار کم روش پیشنهادی می باشد.

همان طور که مشاهده می شود استفاده از هشت سطل باعث کاهش مؤثر تأخیر کل گردیده است بدون اینکه تنش تغییر قابل ملاحظه ای داشته باشد. یعنی می توان یا تأخیر خیلی کمتر نیز گروه چندپخشی را ایجاد نمود. کاهش تأخیر به علت کاهش عمق درخت از \log_4^N به \log_8^N می باشد. افزایش بیشتر تعداد سطل ها اگر چه باز هم باعث کاهش تأخیر می شود، اما تنش را نیز به طور قابل ملاحظه ای افزایش خواهد داد. نمودارهای مربوط به درخت کامل هشت سطلی نیز شبیه شکل پنج هستند و رفتار مشابهی را نشان می دهند. با این تفاوت که تنش Bincast از Nice کمی بهتر است.



شکل ۶- سربار کنترلی روش پیشنهادی

در شکل (۶) مقیاس پذیری دو روش Bincast و Nice مورد مطالعه قرار می گیرد. برای این منظور، تعداد پیغام های ارسال شده بین گره ها برای ساخت و نگهداری درخت در مقیاس لگاریتمی آورده شده است. همان طور که مشاهده می کنید تعداد پیغام های

۵- نتیجه‌گیری

روش‌های سلسله‌مراتبی و براساس درخت به‌خاطر سربرار کنترلی کم و کارایی مناسب بسیار مقیاس‌پذیر می‌باشند [۱۷]. در این مقاله ما یک روش جدید در حیطه ALM ارائه نمودیم که گروه‌های نزدیک به همدیگر را با روش سطل‌بندی در یک سطل قرار می‌دهد و دارای سربرار کنترلی کمی نسبت به 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 Bicast 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، Bicast با حفظ معیار پایداری به تأخیر کمتر و تنش مشابه دست یافته است. ضمن این‌که Bicast با انتخاب پایداری‌ترین 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

- 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. Wu, 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>.
- [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. Bhattacharjee, C. Kommareddy; "Scalable Application Layer Multicast", In Proceeding of ACM Sigcomm, August 2002.
- [3] B. Zhang, S. Jamin, L. Zhang; "Host Multicast: A Framework 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, 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

۷- پی نوشتها

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