

ارائه روشی برای بهبود الگوریتم برش بی‌تی با در نظر گرفتن معیار حافظه

سعیده وثاقتی فاضل*، دانشجوی کارشناسی ارشد مهندسی فناوری اطلاعات، گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه

بوعلی سینا، همدان

Saeede.vesaghati@gmail.com

استاد راهنما: مهدی عباسی استادیارگروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بوعلی سینا

پاییز ۱۳۹۷

چکیده:

یکی از موانع کلیدی برای مسیریاب‌های با سرعت بالا، جستجو برای مسیریابی بسته‌ها می‌باشد. یک بسته ورودی در یک مسیریاب، باید با توجه به قوانین موجود در جدول مسیریابی، جستجویی به منظور ارسال بسته به سمت مقصد انجام دهد. برای هر قانون در جدول مسیریابی، عملی مشخص شده است. در نهایت عمل حاصل شده از نتیجه جستجو، به بسته‌ها اعمال می‌گردد. سرویس‌هایی مانند دیواره‌های آتش، تشخیص نفوذ و ... نیازمند روش‌هایی برای تفکیک بسته‌ها با توجه به قوانین جدول مسیریابی می‌باشند. مجموعه قوانین موجود در جدول مسیریابی را دسته‌بندی می‌نامند و روش‌هایی که به منظور تفکیک بسته استفاده می‌گردد، دسته‌بندی بسته‌ها نامیده می‌شود. دسته‌بندی بسته‌ها یکی از بخش‌های مهم کاربرد اینترنت مانند مسیریابی، مانیتورینگ، امنیت و ... می‌باشد [۱].

واژه‌های کلیدی: مسیریابی، قانون، دسته‌بندی، دسته‌بندی بسته‌ها

۱. مقدمه:

در سال‌های اخیر با افزایش تعداد کاربران شبکه اینترنت و ازدیاد ترافیک موجود در خطوط ارتباطی، میزان تقاضای پهنای باند افزایش یافته است. لذا طراحان شبکه بر آن شدند تا سرعت و کارایی اجزاء مختلف شبکه را بالا ببرند. یکی از راه‌های رسیدن به سرعت بالا، استفاده از سیستم‌های شبکه‌ای است که از دسته‌بندی بسته‌ها به عنوان پردازش پایه‌ای خود استفاده می‌کنند. فرایند طبقه‌بندی بسته‌های شبکه به جریان‌های مختلف، در مسیریاب‌های اینترنت و سوئیچ‌ها را، دسته‌بندی بسته‌ها می‌نامند. الگوریتم‌های دسته‌بندی بسته‌ها را به چهار گروه اصلی جستجوی کامل، تجزیه، فضای چندتایی و درخت تصمیم‌گیری تقسیم کرده است. درخت تصمیم‌گیری، یکی از محبوب‌ترین روش‌های دسته‌بندی بسته‌ها براساس چندین فیلد می‌باشد. تعداد دسترسی‌ها به حافظه و اندازه حافظه مصرفی برای ذخیره داده ساختار درخت، دو معیار کلیدی برای سنجش درخت‌های تصمیم‌گیری می‌باشند. الگوریتم برش بی‌تی، یکی از الگوریتم‌های مبتنی در درخت تصمیم‌گیری می‌باشد که باعث کاهش تعداد دسترسی به حافظه، افزایش کارایی و در نهایت استفاده بهتر از حافظه شده است. در این پژوهش ما به دنبال ارائه روشی کارآمد برای کاهش میزان حافظه مصرفی در عملکرد الگوریتم برش بی‌تی هستیم.

۲. کارهای مرتبط:

از جمله کارهای پیشین در دسته درخت تصمیم‌گیری، می‌توان به موارد زیر اشاره کرد:

۱. الگوریتم برش هوشمندانه سلسله مراتبی

در این الگوریتم، هر گره نشان‌دهنده قسمتی از فضای d بعدی است. بنابراین، در این الگوریتم، گره ریشه کل فضا را نشان می‌دهد. در هر گره یک بعد انتخاب شده و فضا به 2^d قسمت مساوی تقسیم می‌شود. این تقسیم فضا زمانی متوقف می‌شود، که تعداد قوانین موجود در گره، کمتر از یک مقدار از پیش تعریف شده باشد. از آنجایی که این الگوریتم، برای جداسازی، برش‌های مساوی و با توانی از دو انجام می‌دهد، برش‌های ریز برای جداسازی قوانین کوچک، باعث تکرار قوانین بزرگ در بین گره‌ها می‌شوند و در نتیجه میزان حافظه مصرفی بصورت نمائی افزایش می‌یابد. برای جلوگیری از این اتفاق، یک فاکتور از پیش تعریف شده، برای محدود کردن مجموع همه‌ی قوانین موجود در فرزندان گره استفاده می‌شود. برای انتخاب بعد برش نیز، بعدی را انتخاب می‌کند که حداکثر تعداد قوانین فرزندان آن کمتر باشد [۲].

۲. الگوریتم برش‌های چند بعدی

این الگوریتم، بهبود یافته‌ی الگوریتم برش هوشمندانه سلسله مراتبی می‌باشد؛ با این تفاوت که در هر گره، به جای انتخاب فقط یک بعد برای برش، چندین بعد را بطور همزمان برای برش انتخاب می‌کند. هدف از این کار، کاهش عمق درخت و در نتیجه افزایش سرعت دسته‌بندی بسته‌ها به نسبت روش قبلی می‌باشد. در این الگوریتم نیز، همانند روش برش هوشمندانه سلسله‌مراتبی، در هر گره غیر برگ، فضا با برش‌های همزمان به چندین بخش تقسیم شده و از آرایه‌ای از اشاره‌گرها برای اندیس‌گذاری زیر فضا و قوانین موجود در هر یک از آنها به عنوان فرزندان گره برش خورده استفاده می‌شود. در این الگوریتم، با توجه به انتخاب همزمان چندین بعد برای برش، نحوه‌ی انتخاب بعدهای برش و تعداد برش‌ها پیچیده تر است. این الگوریتم برای بهبود حافظه مصرفی، از روش بالاراندن قوانین در درخت تصمیم‌گیری استفاده می‌کند، چرا که تکرار بعضی از قوانین در چندین برگ درخت، موجب افزایش حافظه مصرفی می‌شود. برای حل این مشکل، تمام قوانین مشترک در یک زیر درخت، بصورت یک لیست خطی، در ریشه آن درخت ذخیره می‌شوند [۳].

۳. الگوریتم برش‌های کارا

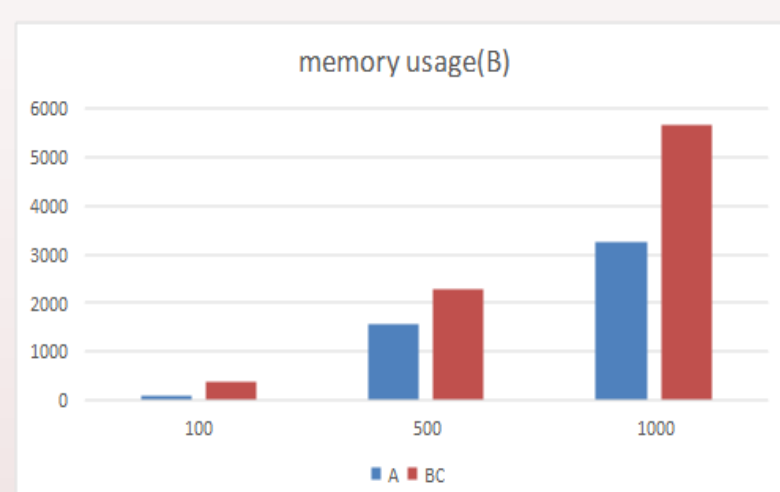
این الگوریتم، بهبود یافته‌ی الگوریتم برش‌های چند بعدی است و برای این کار از گروه‌بندی و برش‌های با تعداد قوانین برابر استفاده می‌کند. علت استفاده از گروه‌بندی، کاهش حافظه مصرفی می‌باشد. منظور از گروه‌بندی، پردازشی است که در آن قوانین هم اندازه را در یک زیرمجموعه قرار می‌دهد تا به راحتی از هم تفکیک شده و تکرار آنها کاهش یابد. ایده استفاده از برش‌های با تراکم برابر، به این صورت است که در نواحی با تراکم بالا از برش‌های کوچک و در نواحی با تراکم پایین از برش‌های بزرگتر استفاده می‌شود. این الگوریتم ممکن است در مقایسه با الگوریتم برش‌های چند بعدی، عمق درخت را افزایش دهد اما قطعاً حافظه مصرفی را کاهش می‌دهد [۴].

۴. الگوریتم برش بی‌تی

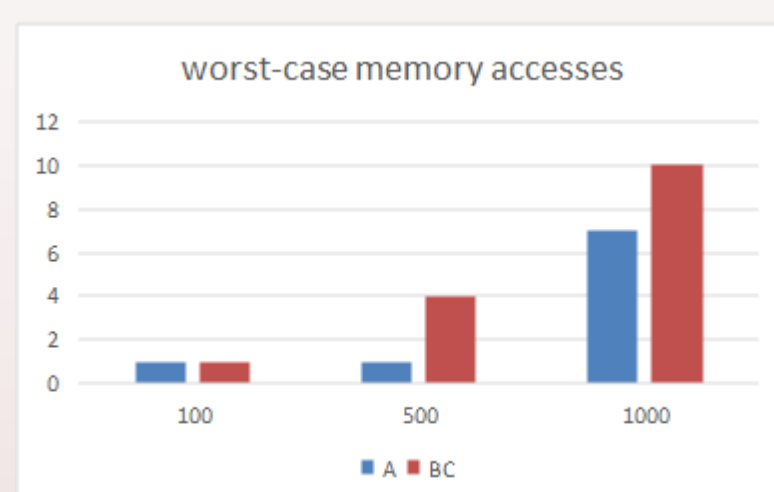
ایده اصلی این الگوریتم، استفاده از بهترین بیت‌های موجود در فیلدهای قوانین برای برش فضا می‌باشد. برای این کار، از یک فرآیند انتخاب بیت استفاده می‌شود، که در هر گره، بیت‌هایی را برای برش انتخاب می‌کند، که منجر به کمترین تکرار قوانین در سطح بعد شود. هدف اصلی این الگوریتم بهبود سرعت دسته‌بندی بسته‌ها و کاهش حافظه مصرفی می‌باشد. در جهت کاهش حافظه، با استفاده از برش بی‌تی که منجر به برش در نواحی پر تراکم می‌شود، سعی در کاهش تعداد گره‌های تهی، تکراری و تک فرزندی دارد. برای تسریع در پیمایش درخت از دستور PEXT استفاده می‌شود که فقط در سه سیکل، با استفاده از فیلد سرآیند بسته و بیت‌های برش انتخابی در گره کنونی، گره بعدی در پیمایش را تعیین می‌کند. این الگوریتم در مقایسه با سه روش قبلی سرعت بهتری کسب و حافظه کمتری مصرف می‌کند [۵].

۳. تحلیل و ارزیابی:

در اکثر روش‌های پیشین، رسیدن به حافظه مصرفی بهینه، منجر به کاهش سرعت جستجو در قوانین می‌شود یا با افزایش سرعت جستجو حافظه مصرفی نیز افزایش می‌یابد. با اینکه الگوریتم برش بی‌تی بصورت قابل ملاحظه‌ای این مشکل را کاهش داده است اما باز هم در نتایج آن هدر رفت حافظه مشاهده می‌شود. ما در این پژوهش تلاش می‌کنیم تا با اصلاح نحوه انتخاب بیت و شرایط اختتام برش در هر گره، هدر رفت حافظه را کاهش دهیم. پیاده سازی با استفاده از زبان C و ابزار class-bench برای تولید مجموعه دسته‌بندی، انجام گرفته است. سه دسته مجموعه قانون تولید شده FW، ACL و IPC با اندازه‌های مختلف از جمله ۱۰۰، ۵۰۰، ۱۰۰۰، ۵۰۰۰ و ۱۰۰۰۰ برای ارزیابی مورد استفاده قرار گرفته‌اند.



شکل ۲: حافظه مصرفی



شکل ۱: تعداد دسترسی به حافظه

۴. نتیجه‌گیری:

نتایج آزمایش نشان می‌دهد که انتخاب بیت هوشمندانه تر در هر گره، منجر به کاهش عمق درخت و در نتیجه افزایش سرعت جستجو می‌شود و از طرفی کاهش حافظه مصرفی را نیز در پی دارد. در ادامه پژوهش ما تلاش می‌کنیم با انتخاب معیار مناسب‌تری برای توقف انتخاب بیت در هر گره، حافظه مصرفی را بهینه تر کنیم.

۵. منابع:

1. Gupta, P. and N. McKeown, Algorithms for packet classification. IEEE Network, 2001. 15(2): p. 24-32.
2. P. Gupta, N. McKeown, Classifying packets with hierarchical intelligent cuttings. IEEE Micro, 2000: p. 34-41
3. S. Singh, F.B., G. Varghese, J. Wang. Packet classification using multidimensional cutting. in Proceedings of the 2003 Conference on Application, Technologies, Architectures, and Protocols for Computer Communications, ACM. 2003.
4. B. Vamanan, G.V., T. Vijaykumar, Efficuts: optimizing packet classification for memory and through. ACM SIGCOMM Computer Communication for Review, 2010. 40: p. 207-218.
5. Liu, Z., et al., BitCuts: A fast packet classification algorithm using bit-level cutting. Computer Communications, 2017. 109: p. 38-52.