

بررسی و مقایسه کارآیی الگوریتم‌های مسیریابی تحمل‌پذیر اشکال در شبکه‌های توری

آرش شماعتی، عباس نایبی*، حمید سربازی‌آزاد[†]

چکیده

مسئله تحمل‌پذیری اشکال در تمامی سیستم‌هایی که در دنیای واقعی عمل می‌کنند از مهمترین مسائل است و در یک سیستم چند پردازنده (بخصوص با تعداد زیاد پردازنده) می‌باید آن را مد نظر قرار داد. الگوریتم‌هایی نیز جهت مسیریابی تحمل‌پذیر اشکال در شبکه‌های میان ارتباطی ارائه شده‌اند. هر کدام از این الگوریتم‌ها نقاط ضعف و قوت خود را دارند. در این مقاله الگوریتم‌ها را از نقطه نظر کارآیی کلی سیستم با یکدیگر مقایسه می‌کنیم. ابزار مقایسه شبیه‌سازی است و نتایج بدست آمده راه‌کارهایی را جهت انتخاب روش مناسب در شرایط مختلف ارائه می‌کند. در این مقاله شبکه توری را بدلیل داشتن خواص ساختاری ویژه، سادگی در پیاده‌سازی، داشتن الگوریتم مسیریابی ساده و کارآیی بالا که آن را مورد توجه سازندگان چندکامپیوترها قرار داده‌است، مد نظر قرار می‌دهیم.

کلمات کلیدی

شبکه‌های میان ارتباطی، مسیریابی تحمل‌پذیر اشکال، ارزیابی کارآیی، شبیه‌سازی

Performance Evaluation of Fault-Tolerant Routing in Mesh Interconnection Networks

A. Shamaei, A. Nayebi, H. Sarbazi-Azad

Sharif University of Technology, Institute for studies in Theoretical Physics and Mathematics

Abstract

Fault-tolerance in the real systems is one of the most important issues especially in multi-computers with a large number of nodes that increases the possibility of fault occurrence in the whole system. There have been many studies focusing on this issue in multi-computers and many researchers have proposed fault-tolerant routing algorithms in multi-computer interconnection networks each of which with some advantages and some disadvantages. In this paper, we evaluate the performance of fault-tolerant routing algorithms. For this purpose we use a discrete-event simulator and to implement the algorithms and evaluate their performance. The network used in this study is the famous mesh which is used in many multi-computers due to its desirable properties such as simplicity, regular structure, simple routing algorithm and ability to utilize locality in traffic loads.

Keywords

Interconnection Network, Fault-tolerant routing, Performance Evaluation, Simulation.

* دانشگاه صنعتی شریف، تهران، میدان آزادی، خیابان آزادی a_shamaei@ce.sharif.edu, nayebi@ce.sharif.edu

[†] عضو هیأت علمی دانشگاه صنعتی شریف، دانشکده مهندسی کامپیوتر، پژوهشگاه دانشهای بنیادی، پژوهشکده علوم کامپیوتر azad@ipm.ir

۱- مقدمه

بر الگوریتم‌هایی که در این زمینه ارائه شده‌اند. در بخش چهارم الگوریتم‌هایی که مورد بررسی دقیق قرار گرفته‌اند، با هم مقایسه می‌شوند و در بخش آخر نتیجه‌گیری و کارهای آینده آورده می‌شود.

۲- مفاهیم اولیه تحمل پذیری اشکال

خواص و تعاریف زیر برای شبکه‌های میان‌ارتباطی قابل تعریف و بررسی هستند [3].

- شبکه متصل: یک شبکه میان‌ارتباطی با توجه به یک الگوریتم مسیریابی متصل نامیده می‌شود اگر تابع مسیریابی بتواند یک پیام را بین هر دو گره بدون اشکال مسیریابی کند.
- کانال تکراری^۳: یک کانال تکراری نامیده می‌شود اگر پس از حذف آن شبکه کماکان متصل و بدون بن‌بست بماند.
- f-اشکال پذیر^۴: یک الگوریتم مسیریابی f-اشکال پذیر نامیده می‌شود اگر هر f عنصر از شبکه دچار اشکال شدند شبکه کماکان متصل و بدون بن‌بست بماند.
- f-اشکال باز یافتنی^۵: یک شبکه f-اشکال باز یافتنی است اگر هر f عنصری از شبکه دچار اشکال شدند پیام‌های به اشکال خورده نتوانند منابع شبکه را برای مدت نامحدود در اختیار بگیرند. باید توجه داشت که این شرط از شرط f-اشکال پذیری ضعیفتر است.

۲-۱- کلاسهای خرابی

- **خرابی گره^۶**: خرابی عنصر پردازشی یا مسیریاب یا خرابی‌های نرم‌افزاری در لایه تبادل پیام^۷ در این کلاس از خرابی‌ها قرار می‌گیرند.
- **خرابی کانال^۸**: خرابی کنترل کننده کانال یا خرابی کانال فیزیکی یا خرابی بافرهای کانال‌های مجازی از این کلاس است.

۲-۲- نواحی اشکال^۹

گره‌ها و کانال‌های اشکال دار مجاور هم تشکیل نواحی اشکال را می‌دهند. این نواحی را می‌توان از نظر شکل به دو دسته تقسیم کرد:

- **نواحی محدب^{۱۰}**: کار با این نواحی معمولاً ساده‌تر است. یک نمونه خاص از این نوع نواحی، نواحی بلوکی هستند. این نواحی بطور طبیعی در شبکه توری در صورت خرابی یک برد یا یک پیمانانه بوجود می‌آیند.
- **نواحی مقعر^{۱۱}**: بررسی این نواحی معمولاً مشکل‌تر است و الگوریتم‌های مسیریابی که می‌توانند با این نوع اشکال‌ها کار کنند نیز محدود و پیچیده هستند. برای اینکه از الگوریتم‌های ساده و متنوعی که فقط با نواحی محدب یا بطور خاص بلوکی کار می‌کنند در اینجا نیز بتوان استفاده کرد این نواحی باید به نواحی محدب یا بلوکی تبدیل شوند. برای این کار لازم است که بعضی از گره‌های سالم شبکه به عنوان گره اشکال دار علامت گذاری شوند.

همزمان با پیدایش کامپیوتر، تکاپوی بشر برای رسیدن به قدرت پردازشی بیشتر فزونی یافت و در حال حاضر در عرصه‌هایی نظیر ژنتیک، هوافضا، فیزیک اتمی و هواشناسی به یک نیاز جدی مبدل شده‌است. متخصصان قصد داشته و دارند با استفاده از هر ابزاری به این هدف برسند و راه‌های متعددی نیز برای رسیدن به این هدف ارائه کرده‌اند که از جمله آنها می‌توان به، بکارگیری تکنولوژی VLSI، تسریع با استفاده از تکنیک‌هایی در سطح معماری پردازنده (نظیر ساخت پردازنده‌های خاص منظوره^۱، برداری، فوق‌عددی و...)، بهبود کد برنامه‌ها و بهینه‌کردن آنها و استفاده همزمان از چندین پردازنده برای انجام وظایف اشاره کرد.

همانطور که گفتیم یکی از راه‌های رسیدن به قدرت پردازشی بیشتر بهره بردن از پردازش موازی و بکار گرفتن همزمان چند پردازنده می‌باشد. اما مسائلی در این میان پیش می‌آیند، که بررسی آنها منجر به پیدایش زمینه‌ای بنام شبکه‌های میان‌ارتباطی^۲ شده‌است. این زمینه به بررسی مسائلی نظیر نحوه اتصال پردازنده‌ها از لحاظ فیزیکی، طرز ارسال پیام بین این عناصر پردازشی، الگوریتم مسیریابی (که بسته را از مبدأ به مقصد می‌رساند) می‌پردازد.

مهمترین وجه تمایز شبکه‌های میان‌ارتباطی با سایر شبکه‌های معمول در نظم بالای این شبکه‌هاست. منظم بودن این شبکه‌ها باعث می‌شود تا الگوریتم‌های مسیریابی در این شبکه‌ها بسیار ساده باشند و سرعت انتقال اطلاعات در این شبکه‌ها بالا باشد. همچنین تاخیر انتقال بسته‌ها کم و در شرایط عادی تا حدودی ثابت است.

احتمال بروز اشکال در شبکه‌های میان‌ارتباطی یعنی احتمال از بین رفتن نظم موجود در این شبکه‌ها، و این درست به معنی پیچیدگی مسیرها و الگوریتم‌های مسیریابی است زیرا هرچقدر مسیرها پیچیده‌تر شود گراف وابستگی پیچیده‌تر شده و احتمال بروز بن‌بست بیشتر می‌شود و جلوگیری از آن مشکل‌تر خواهد بود [1].

لازم به ذکر است که در سیستم‌های فعلی احتمال بروز اشکال در مسیریاب‌ها بسیار کم است. با وجود این در بعضی از کاربردهای حساس نظیر امور نظامی و فضایی نمی‌توان این مسئله را نادیده گرفت زیرا در صورت بروز اشکال زمان لازم برای اصلاح آن بسیار زیاد است، و حتی گاهی نیز اصلاح اشکال ناممکن است. از طرف دیگر با بالا رفتن تعداد عناصر در یک شبکه احتمال بروز حداقل یک اشکال در شبکه بالا می‌رود [2].

تکنیک‌هایی که در این مقاله به آنها اشاره می‌شود از مهمترین و پایه‌ای‌ترین تکنیک‌های مسیریابی تحمل‌پذیر اشکال در شبکه‌های میان‌ارتباطی هستند که می‌توانند در بسیاری از زمینه‌های مشابه نیز به کار گرفته شوند. در بخش دوم این مقاله مفاهیم اولیه تحمل‌پذیری اشکال معرفی می‌شوند. از آنجا که بررسی الگوریتم‌های تحمل‌پذیر اشکال هدف این مقاله می‌باشد، در قسمت سوم مروری خواهیم داشت

۲-۳- حالات اشکال

اشکال‌ها را می‌توان به دو دسته تقسیم کرد:

- **اشکال‌های ایستا^{۱۲}**: به هنگام روشن شدن سیستم ظاهر می‌شوند.
 - **اشکال‌های پویا^{۱۳}**: به صورت پویا در جریان کار شبکه ایجاد می‌شوند.
- همچنین اشکال‌ها را می‌توان از نظر رفتار گذرا به دو دسته تقسیم کرد
- **اشکال‌های مانا^{۱۴}**: این اشکال‌ها پس از ایجاد شدن در سیستم باقی می‌مانند تا زمانی که تعمیر شوند.
 - **اشکال‌های گذرا^{۱۵}**: این اشکال‌ها در سیستم ایجاد می‌شوند و بعد از مدتی از بین می‌روند. در مورد این اشکال‌ها باید دقت شود که پس از برطرف شدن آثار دائمی از خود بر جای نگذارند.

۳- الگوریتم‌های مسیریابی تحمل‌پذیر اشکال

قبل از استفاده از روش راهگزینی خزشی بیشتر الگوریتم‌ها از روش راهگزینی بسته‌ای یا روش VCT^{۱۶} استفاده می‌کردند. البته الگوریتم‌هایی نیز ارائه شده‌اند که از روش PCS^{۱۷} استفاده می‌کردند (Yalamanchili و Gaughan از این روش استفاده کرده‌اند [4]). اکثر الگوریتم‌هایی که از روش راهگزینی خزشی استفاده می‌کنند بر پایه اضافه کردن کانال‌های مجازی به کانال فیزیکی می‌باشند. اضافه کردن بافرهای خاص و مدارات کنترلی به یک کانال فیزیکی این امکان را می‌دهد که به صورت طبیعی یک بسته را در هر زمان منتقل کند در حالیکه این زمان بین پیام‌های متفاوت مشترک می‌باشد. بنابراین هر کانال فیزیکی تعدادی کانال مجازی خواهد داشت.

کانال‌های مجازی در ابتدا به منظور جلوگیری از بن‌بست و پشتیبانی مسیریابی تطبیقی استفاده می‌شدند و فقط به منظور تحمل‌پذیری بودن اشکال به کار نمی‌رفتند. برای مثال Dally و Seitz از کانال‌های مجازی برای مسیریابی غیرتطبیقی استفاده می‌کردند که تحمل‌پذیر اشکال نیز نبود [5]. بعضی از الگوریتم‌های کمینه نیز از کانال‌های مجازی استفاده می‌کردند، اما مسیریابی‌های کمینه اصولاً تحمل‌پذیر اشکال نمی‌باشند. برای تحمل‌پذیری اشکال معمولاً باید بتوان مسیری غیر از مسیر کمینه را انتخاب کرد.

تحقیقات زیادی صورت گرفته که در آنها از کانال مجازی برای طراحی الگوریتم‌های تحمل‌پذیر اشکال استفاده شده است. یکی از ابتدایی‌ترین این تحقیقات الگوریتمی است که توسط Linder و Harden ارائه شد [6]. روش این دو برای طراحی یک الگوریتم تطبیقی تحمل‌پذیر اشکال، ایجاد شبکه‌های مجازی^{۱۸} برای ارسال در هر جهت در شبکه توری می‌باشد. برای این که این الگوریتم بتواند حداقل یک اشکال را تحمل کند، باید تعداد کانال‌های مجازی مورد استفاده در هر بعد دو برابر شود. این کانال‌های اضافه منحصراً برای مسیریابی حول ناحیه اشکال به کار می‌روند. برای استفاده از این

الگوریتم در یک شبکه توری n بعدی به 2^n کانال مجازی برای هر کانال فیزیکی در دو بعد و 2^{n-1} کانال مجازی برای هر کانال فیزیکی در سایر ابعاد احتیاج است. یکی از نقاط ضعف این روش به کارگیری تعداد بسیار زیاد کانال‌های مجازی می‌باشد. دیگر نقطه ضعف این الگوریتم این است که با وجود استفاده از کانال‌های مجازی بسیار، این الگوریتم به حد کافی تطبیقی و تحمل‌پذیر اشکال نمی‌باشد.

Chien و Kim نیز تلاش کردند تا از کانال‌های مجازی برای طراحی الگوریتم‌های تحمل‌پذیر اشکال در شبکه‌های توری استفاده کنند [7]. آنها ابعاد شبکه توری را رتبه‌بندی کردند و برای هر ترتیب ابعاد یک شبکه مجازی ساختند و به پیام‌ها اجازه می‌دادند تا از یک شبکه مجازی به سوی شبکه مجازی دیگر پیشروی داشته باشند. نتیجه کار آنها یک الگوریتم تطبیقی غیر کمینه^{۱۹} بود که می‌توانست پیام‌ها را در هر زمانی در دو بعد منتقل کند. مزیت عمده این روش نسبت به روش Linder و Harden صرفنظر از تعداد ابعاد، استفاده از سه کانال مجازی برای هر کانال فیزیکی می‌باشد. نقطه ضعف اصلی این روش این است که اشکال در گره‌های مرزی نمی‌توانند واقع شوند.

Duato یکی از موفق‌ترین الگوریتم‌ها را برای تحمل‌پذیری اشکال تا زمان خود ارائه کرد [8]. او پنج شبکه مجازی برای پیام‌ها به وجود آورد. یکی برای مسیریابی پیام‌هایی که از ابتدا به سوی شرق می‌روند، یکی برای مسیریابی پیام‌هایی که از ابتدا به سوی غرب می‌روند، یکی برای مسیریابی پیام‌هایی که در مرز شرقی شبکه توری با اشکال مواجه می‌شوند، یکی برای مسیریابی پیام‌هایی که در مرز غربی شبکه توری با اشکال مواجه می‌شوند و یکی برای مسیریابی پیام‌ها در هر لحظه. نتیجه این روش، الگوریتمی بود که می‌توانست حداقل $n-1$ اشکال را در یک شبکه n بعدی تحمل کند. این الگوریتم به چهار کانال مجازی برای هر کانال فیزیکی احتیاج داشت. با اینکه این روش نسبت به روش قبلی یک کانال مجازی بیشتر داشت، اما در کل نسبت به دو الگوریتم قبلی هم تطبیقی تر بود و هم قابلیت تحمل‌پذیری اشکال بیشتری داشت. یکی دیگر از مزایای این روش نسبت به دو روش قبلی این بود که می‌توانست اشکال‌های پویا (اشکال‌هایی که در زمان کار سیستم رخ می‌دهند) را نیز تحمل کند. حال اینکه دو روش قبلی فقط می‌توانستند اشکال‌های ایستا را تحمل کنند. تحمل کردن اشکال‌های ایستا در واقع زیاد عملی نیست، زیرا گره‌ها و کانال‌ها می‌توانند در هر لحظه معیوب شوند و زیاد خوشایند نیست که سیستم بعد از هر اشکال و قبل از هر تعمیر خاموش شود.

Boppana و Chalasani الگوریتم تحمل‌پذیر اشکالی ارائه دادند که از بن‌بست به کمک استفاده از چهار کانال مجازی برای هر کانال فیزیکی جلوگیری می‌کرد [9]. در این الگوریتم حول نواحی اشکال حلقه‌ای تشکیل می‌شود و پیام‌ها در صورت برخورد با ناحیه اشکال در این حلقه به مسیریابی منحرف می‌شوند. یکی از مشکلات اساسی این الگوریتم کار کردن با نواحی اشکال به صورت مستطیلی بود که همین دو نفر در سال ۱۹۹۷ الگوریتم خود را طوری بهبود دادند که بتواند

علاوه بر تحمل اشکال‌های مستطیلی شکل، اشکال‌های خاصی بصورت $T, L, +$ را نیز تحمل کند [10]. اما Sui و Wang نشان دادند که الگوریتم [9] در یک حالت خاصی با بن‌بست مواجه می‌شود، بنابراین الگوریتم قبلی را طوری بهبود دادند که علاوه بر حل مشکل مذکور، از یک کانال مجازی کمتر نسبت به قبلی نیز استفاده می‌کرد [11].

Boura و Das در سال ۱۹۹۵ مفهومی به نام نامگذاری گره‌ها به سه حالت سالم، اشکال‌دار و ناامن^{۲۰} را معرفی کردند [12]. الگوریتم این دو برای شبکه‌های توری دو بعدی بود که از سه کانال مجازی استفاده می‌کرد. مسیریابی به صورت کاملاً تطبیقی در نواحی سالم شبکه انجام می‌شد و در صورت برخورد با نواحی اشکال، پیام حول ناحیه اشکال منحرف می‌شد. اشکال این الگوریتم نیز همانند الگوریتم‌های قبلی این است که برای شکل‌گیری ناحیه اشکال مستطیلی می‌باید تعدادی از گره‌های سالم اشکال‌دار فرض شوند.

Ni و Glass در سال ۱۹۹۶ الگوریتمی ارائه دادند که بر اساس مدل چرخشی^{۲۱} استوار بود [13]. این الگوریتم از کانال مجازی استفاده نمی‌کند، اما محدودیت‌های زیادی دارد. این الگوریتم در یک شبکه توری دو بعدی فقط می‌تواند یک اشکال را تحمل کند و به طور کلی $n-1$ اشکال پذیر می‌باشد.

Shin و Su یک الگوریتم تحمل‌پذیر اشکال برای شبکه‌های توری n بعدی ارائه دادند [14]. این الگوریتم می‌توانست اشکال‌های مستطیلی غیرمتصل را تحمل کند، به شرطی که فاصله هر دو گره اشکال‌دار در هر دو بلوک اشکال در هر بعد حداقل سه باشد. Shih الگوریتم ارائه شده قبلی را بهبود بخشید به طوری که فاصله گره‌های اشکال‌دار در هر بلوک متفاوت اشکال را به دو رساند [15].

Wu در سال ۲۰۰۰ روشی برپایه اطلاعات سراسری محدود برای الگوریتم‌های تحمل‌پذیر اشکال ارائه داد [16]. این روش نیازی به کانال مجازی نداشت اما محدودیت‌های خاص خود را داشت. نواحی اشکال باید مستطیلی شکل باشند و همچنین فاصله این نواحی از هم باید حداقل ۳ باشد. همین شخص در سال ۲۰۰۳ الگوریتم دیگری ارائه داد که باز هم به کانال مجازی نیازی نداشت ولی این‌بار اطلاعات سراسری کمتری نیاز داشت [17]. این روش بر پایه مدل چرخشی زوج و فرد می‌باشد که در فصل بعد مورد بررسی قرار خواهد گرفت.

Chiu و Chen در سال ۲۰۰۱ الگوریتمی ارائه دادند که در واقع مکمل روش Boppana و Chalasani می‌باشد [18]. این الگوریتم قطعی بوده و از ۳ کانال مجازی سود می‌برد. مزیت عمده این روش، تحمل اشکال در نواحی اشکال غیر محدب می‌باشد.

Zhou و Lau الگوریتم‌هایی ارائه داده‌اند که حول نواحی اشکال حلقه تشکیل داده و پیام در صورت برخورد با ناحیه اشکال، در طی این حلقه اشکال منحرف می‌شوند [19][20]. الگوریتم‌های ارائه شده تطبیقی می‌باشند که تفاوت آنها در تعداد کانال‌های مجازی مورد استفاده می‌باشد. اولین الگوریتم از ۳ کانال مجازی و دومی از ۲ کانال

مجازی استفاده می‌کند. هر دوی این الگوریتم‌ها با این نقطه ضعف مواجه‌اند که مجبورند تعدادی از گره‌های سالم را قربانی کنند.

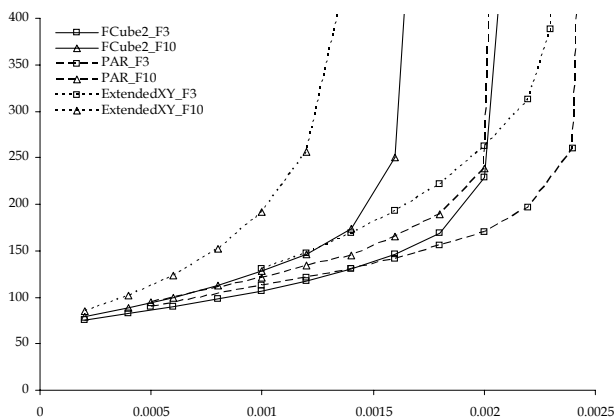
Ho و Stockmeyer ایده به کارگیری گره‌های λ را معرفی کردند [21]. گره‌های λ گره‌هایی هستند که سالم بوده ولی قدرت پردازشی ندارند. در واقع این گره‌ها قربانی می‌شوند. گره‌های λ طوری انتخاب می‌شوند که هر گره سالمی بتواند با دو مرحله از الگوریتم معروف e -cube به هر گره سالم دیگری برسد. الگوریتم مسیریابی قطعی بوده و این روش به ۲ کانال مجازی نیاز دارد.

Chen و Xiang یک الگوریتم تحمل‌پذیر اشکال در شبکه‌های توری دو بعدی ارائه دادند که بر پایه امنیت محلی^{۲۲} استوار بود [22]. مزیت عمده این روش نسبت به روش‌های ارائه شده قبلی این است که لازم نیست گره سالمی را برای شکل‌گیری ناحیه اشکال، اشکال‌دار در نظر گرفت. همچنین Gomez با استفاده از مفهوم گره میانی، بدون اشکال‌دار در نظر گرفتن هیچ گره سالمی، روشی جهت تحمل‌پذیری اشکال در شبکه‌های توری سه بعدی ابداع کرد [23] که این روش گرچه بر پایه اطلاعات سراسری می‌باشد، اما از بقیه روش‌هایی که تا کنون ارائه شده‌اند بهتر عمل می‌کند. Gomez به همراه دیگر همکارانش روش خود را در [2] بهبود داده است.

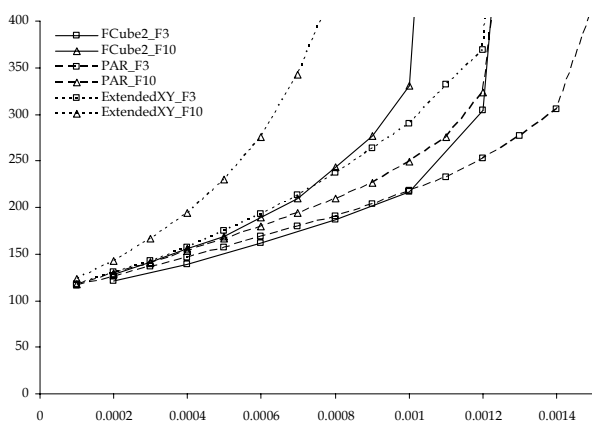
۴- مقایسه الگوریتم‌ها

در این قسمت سه الگوریتم مختلف را برای شبیه‌سازی انتخاب کرده‌ایم. الگوریتم اول، الگوریتم صفحه‌ای^{۲۳} [7] است. از آنجایی که در بیشتر مقالات مورد بحث در این زمینه، از این الگوریتم یاد شده و نویسندگان مقالات روش‌های خود را با این روش مقایسه کرده‌اند، ما این الگوریتم را برای شبیه‌سازی انتخاب کردیم. الگوریتم بعدی الگوریتم $fcube_2$ است که توسط Boppana و Chalasani ارائه شد [9]. این دو نفر اولین نفراتی بودند که از مفهوم حلقه اشکال و زنجیر اشکال در روش خود استفاده کردند. سومین الگوریتمی که برای مطالعه در این مقاله انتخاب شده، الگوریتم XY توسعه یافته می‌باشد [1]. این الگوریتم توسط Wu در سال ۲۰۰۳ ارائه شد. این روش از معدود روش‌هایی است که از کانال مجازی استفاده نمی‌کند و بر پایه اطلاعات محلی استوار است.

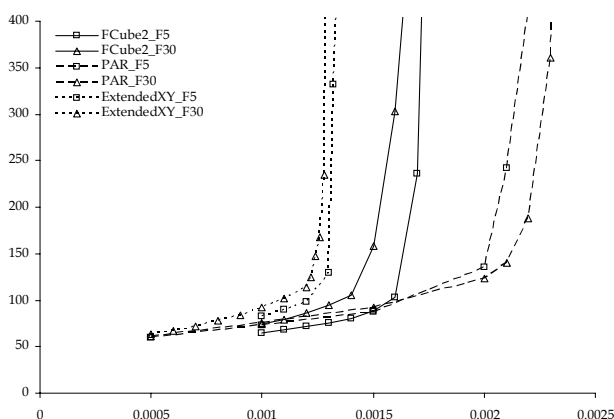
شبیه‌ساز مورد استفاده در این مقاله XMulator^{۲۴} نام دارد [24]. این شبیه‌ساز که نام آن مخفف دو کلمه XML و Simulator می‌باشد، با روش مبتنی بر رویداد کار می‌کند. هر سه الگوریتمی که برای شبیه‌سازی انتخاب شده‌اند، فقط قادرند نواحی اشکال از نوع بلوکی را تحمل کنند. همچنین اشکال در گره‌های مرزی شبکه نمی‌تواند واقع شود. از آنجا که تعداد کانال‌های مجازی مورد استفاده برای هر الگوریتم متفاوت می‌باشد، ما برای اینکه مقایسه یکسانی داشته باشیم، سعی کرده‌ایم تا برای همه الگوریتم‌ها تعداد مساوی از کانال‌های مجازی را انتخاب کنیم. برای این منظور ما هر الگوریتم را با ۶ عدد کانال مجازی شبیه‌سازی کرده‌ایم. الگوریتم $fcube_2$ به دو کلاس



شکل ۲. طول پیام ۶۴ فلپت، اندازه شبکه ۱۰×۱۰



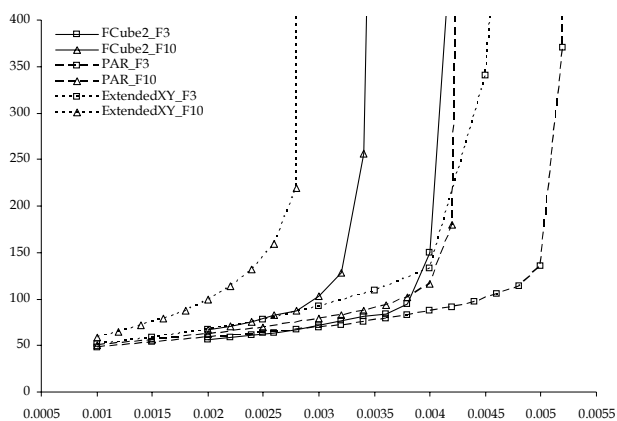
شکل ۳. طول پیام ۱۰۰ فلپت، اندازه شبکه ۱۰×۱۰



شکل ۴. طول پیام ۳۲ فلپت، اندازه شبکه ۲۰×۲۰

از کانال‌های مجازی نیاز دارد، بنابراین برای هر کلاس آن سه کانال مجازی در نظر گرفته شده است. به هرکدام از سه کلاس کانال‌های مجازی الگوریتم‌های صفحه‌ای دو عدد کانال مجازی اختصاص داده شده است. الگوریتم مسیریابی XY توسعه یافته که به کانال مجازی نیاز ندارد با همان ۶ کانال مجازی شبیه‌سازی شده است.

نرخ تولید پیام^{۲۵} در گره‌ها برابر بوده و همچنین توزیع مقصد^{۲۶} پیام‌ها نیز به صورت یکنواخت در نظر گرفته شده است. شبیه‌سازی برای دو اندازه مختلف شبکه ۱۰×۱۰ و ۲۰×۲۰ انجام شده است. در شبکه‌های ۱۰×۱۰ هر الگوریتم یکبار با ۳ و بار دیگر با ۱۰ گره اشکال-دار شبیه‌سازی شده است. در شبکه‌های ۲۰×۲۰ نیز هر الگوریتم یکبار با ۵ و بار دیگر با ۳۰ گره اشکال‌دار شبیه‌سازی شده است. طول پیام در شبیه‌سازیها ۳۲، ۶۴ و ۱۰۰ فلپت می‌باشد. لازم به ذکر است چون الگوریتم XY توسعه یافته فقط با مدل خرابی گره کار می‌کند، ما برای سادگی تمامی الگوریتم‌ها را با مدل خرابی گره شبیه‌سازی کرده‌ایم. خرابی‌های گره به صورت تصادفی در شبکه به وجود آمده‌اند. این خرابی‌ها طوری در نظر گرفته شده‌اند که شامل بلوک‌های اشکال غیرهم‌پوشان باشند. در ادامه نمودارهایی که از شبیه‌سازی بدست آمده‌اند را ملاحظه می‌کنید. محور افقی در تمامی این نمودارها نرخ تولید پیام در هر گره و محور عمودی متوسط تاخیر پیام^{۲۷} در شبکه را نشان می‌دهد.



شکل ۱. طول پیام ۳۲ فلپت، اندازه شبکه ۱۰×۱۰

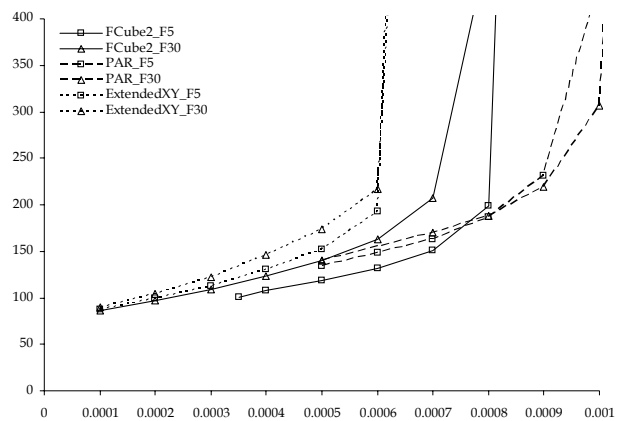
دارد و همچنین با زیاد شدن تعداد اشکالها کارایی آن زیاد افت نمی‌کند. این امر تا حدی طبیعی به نظر می‌رسد زیرا روش مسیریابی در این الگوریتم تطبیقی جزئی می‌باشد، حال اینکه در دو الگوریتم دیگر مسیریابی قطعی است. در الگوریتم XY توسعه یافته وقتی تعداد اشکالها زیاد می‌شود، کارایی شبکه افت بسیار می‌کند و همانطور که در نمودارها مشاهده می‌شود، این الگوریتم با تعداد ۱۰ اشکال نسبت به تمامی الگوریتم‌های دیگر زودتر به اشباع می‌رسد. اما اگر طول پیام و همچنین تعداد اشکال کم باشد، کارایی آن نسبت به fcube-2 بهتر است. در هر صورت الگوریتم XY توسعه یافته محدودیت‌های فراوانی دارد و تنها مزیت آن عدم نیاز به کانال مجازی می‌باشد که این امر موجب ساده بودن مسیریاب‌ها می‌شود. با توجه به شکل‌های ۴ تا ۶ که همین مجموعه یک برای شبکه بزرگتر را نشان می‌دهد درمی‌یابیم که الگوریتم fcube-2 از XY توسعه یافته عملکرد بهتری دارد. اما در هر صورت الگوریتم صفحه‌ای از هر دوی این الگوریتم‌ها بهتر عمل می‌کند.

۵- نتیجه‌گیری

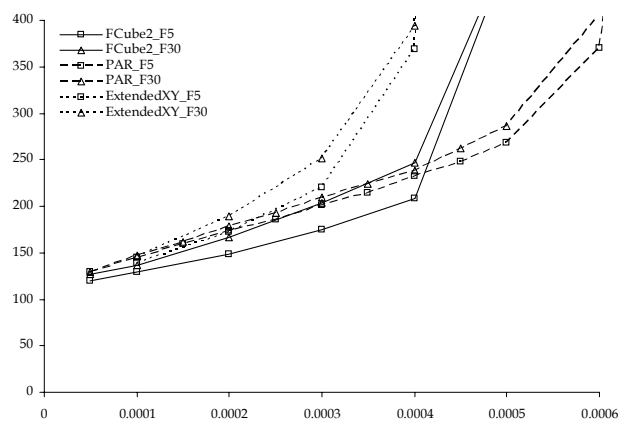
در این مقاله کارایی الگوریتم‌های تحمل‌پذیر اشکال را مقایسه و بررسی کردیم. برای انجام مقایسه بهتر الگوریتم‌ها را تحت شرایط یکسان از نظر تعداد کانال مجازی، ناحیه اشکال، و تعداد گره‌های اشکال دار با هم مقایسه کردیم. شبیه سازی‌ها را برای دو اندازه مختلف شبکه و هر کدام را با سه طول پیام مختلف ۳۲، ۶۴ و ۱۰۰ فلیت تکرار کردیم. تعداد اشکال‌ها را نیز برای هر سناریو دو مقدار مختلف در نظر گرفتیم. دیدیم که هر الگوریتم با توانایی خاص خود می‌توانست در شرایطی بهتر عمل کند.

مراجع

- [1] J. Wu, "A fault tolerant adaptive and minimal scheme in n-d meshes", the computer Journal, Vol. 45, No. 3, pp.349-363, 2002.
- [2] M.E. Gomez, N.A. Nordbotten, J. Flich, P. Lopez, A. Rolbles, J. Duato, T. Skeie & O. Lysne, "A routing methodology for achieving fault tolerance in direct networks," IEEE Trans. On Computers, Vol. 55, No. 4, pp. 400-414, April 2006.
- [3] J. Duato, S. Yalamanchili & L. Ni, "Interconnection Networks: An Engineering Approach", IEEE Computer Society Press, 2003.
- [4] P.T. Gaughan & S. Yalamanchili, "Pipelined Circuit-Switching: A Fault-tolerant Variant of Wormhole Routing," Proc. Fourth IEEE Int'l Symp. Parallel and Distributed Processing, pp. 148-155, Dec 1992.
- [5] W.J. Dally & C.L Seitz, "Deadlock Free Message Routing in Multiprocessor Interconnection Network," IEEE Trans. Computers, vol.36, no. 5, pp. 547-533, May 1987.
- [6] D.H. Linder & J.C. Harden, "An Adaptive and Fault Tolerant Wormhole Routing Strategy for k-Ary n-Cubes," IEEE Trans, Computers, vol.40, no.1, pp.2-12, Jan.1991.
- [7] A.A. Chien & J.M. Kim, "Planer-Adaptive Routing: Low-Cost Adaptive Networks for Multiprocessors," Proc. 19th



شکل ۵. طول پیام ۶۴ فلیت، اندازه شبکه ۲۰×۲۰



شکل ۶. طول پیام ۱۰۰ فلیت، اندازه شبکه ۲۰×۲۰

همانطور که ذکر شد معیار مقایسه در این قسمت متوسط تاخیر پیام می‌باشد که همان محور عمودی در نمودارهای فوق می‌باشد. تاخیر پیام^{۲۸} برابر است با مقدار زمانی که طول می‌کشد یک پیام از مبدا به مقصد برسد. اگر برای تمامی پیام‌هایی که در شبکه تولید می‌شود این مقدار را حساب کنیم و مقدار میانگین آنرا بدست آوریم به عدد متوسط تاخیر پیام می‌رسیم. همانطور که در شکل‌های فوق ملاحظه می‌کنید، متوسط تولید پیام برای هر الگوریتم متناسب با نرخ تولید پیام در ابتدا به صورت خطی افزایش می‌یابد، اما بعد از مدتی این افزایش دیگر خطی نیست و تقریباً به صورت نمایی افزایش می‌یابد. در این حالت است که اصطلاحاً می‌گوییم شبکه به حالت اشباع^{۲۹} رسیده است و شبکه در این حالت دیگر کارآمد نمی‌باشد. بنابراین الگوریتمی نسبت به الگوریتم دیگر برتری دارد که دیرتر وارد ناحیه اشباع خود شود و در ضمن متوسط تاخیر پیام برای آن الگوریتم در یک نرخ تولید پیام نسبت به الگوریتم دیگر کم‌تر باشد. با توجه به شکل‌های ۱ تا ۳ می‌توان دریافت که الگوریتم صفحه‌ای نسبت به دو الگوریتم fcube-2 و XY توسعه یافته عملکرد بهتری

1 ASIC
 2 Interconnection Network
 3 Redundant
 4 f-fault tolerant
 5 f-fault recoverable
 6 Node failure
 7 Messaging layer
 8 Link failure
 9 Fault regions
 10 Convex
 11 Concave
 12 Static
 13 Dynamic
 14 Permanent
 15 Transient
 16 Virtual Cut Through
 17 Pipeline Circuit Switching
 18 Virtual Network
 19 Non minimal
 20 Unsafe
 21 Turn Model
 22 Local Safety
 23 Planar
 24 <http://www.XMulator.org>
 25 Generation Rate
 26 Destination Distribution
 27 Average Message Latency
 28 Message Latency
 29 Saturation

- Ann. Int'l Symp. Computer Architecture, pp.268-277, May 1992.
- [8] J. Duato, "A Theory of Fault-Tolerant Routing in Wormhole Networks," Proc.1994 Int'l Conf. Parallel and Distributed Systems, pp.600-607, Dec.1994.
- [9] R.V. Boppana & S. Chalasani, "Fault-tolerant wormhole routing algorithms for mesh networks," IEEE Trans. Computers, 44(7): 848-864, July 1995.
- [10] S. Chalasani & R.V. Boppana, "Communication in multicomputers with nonconvex faults," IEEE Trans. Computers, 46(5): 616-622, May 1997.
- [11] P.H. Sui & S.D. Wang, "An improved algorithm for Fault-tolerant wormhole routing in meshes", IEEE Trans. Computers, Vol. 46, No. 9, pp. 1040-1042, Sep 1997.
- [12] Y.M. Boura & C.R. Das, "Fault-tolerant routing in mesh networks", Proc. Int. Conf. Parallel Computing, pp. I106-I109, Aug 1995.
- [13] C. Glass & L.M. Ni, "Fault tolerant wormhole routing in meshes without virtual channels", IEEE Trans. On Parallel and Distributed Systems, Vol. 7, No. 6, pp.620-636, June 1996.
- [14] C.C. Su & K.G. Shin, "Adaptive fault-tolerant deadlock-free routing in meshes and hypercubes," IEEE Trans. Computers, 45(6):pp. 666-683, June 1996.
- [15] J.D. Shih, "Adaptive fault-tolerant wormhole routing algorithms for hypercube and mesh interconnection networks," Proc. 11th Parallel Processing Symposium, pp. 333-340, April 1997.
- [16] J. Wu, "Fault-tolerant adaptive and minimal routing in mesh-connected multicomputers using extended safety levels", IEEE Trans. On Parallel and Distributed Systems, Vol. 11, No. 2, pp. 149-159, Feb 2000.
- [17] J. Wu, "A Fault-tolerant and deadlock-free routing protocol in 2D meshes based on Odd-Even Turn model", IEEE Trans. Computers, Vol. 52, No. 9, pp. 1154-1169, Sep 2003.
- [18] C.L. Chen & G.M. Chiu, "A Fault-tolerant routing scheme for meshes with nonconvex faults", IEEE Trans. on Parallel and Distributed Systems, Vol. 12, No. 5, pp. 467-475, May 2001.
- [19] J. Zhou & F.C.M. Lau, "Adaptive Fault-tolerant wormhole routing in 2D meshes", Proc. of 15th Int. Symposium Parallel and Distributed Processing, pp. 278-287, 2001.
- [20] J. Zhou & F.C.M. Lau, "Adaptive fault-tolerant wormhole routing with two virtual channels in 2D meshes", Proc. of 7th Int. Symposium Parallel Architectures, Algorithms and Networks, pp. 142-148, 2004.
- [21] C.T. Ho & L. Stockmeyer, "A new approach to fault-tolerant wormhole routing for mesh-connected parallel computers", IEEE Trans. On Computers, Vol. 53, No. 4, pp. 427-438, Apr 2004.
- [22] D.Xiang & A. Chen, "Fault Tolerant routing in 2D tori or meshes using limited-global-safety information," Proc. Of IEEE Int. Conf. on Parallel Processing, pp.231-238, Vancouver, Aug 2002.
- [23] M.E. Gomez, J. Flich, P. Lopez, A. Rolbles, J. Duato, N.A. Nordbotten, O. Lysne, & T. Skeie, "An efficient fault-tolerant routing methodology for direct networks," Proc. Of IEEE Int. Conf. on Parallel Processing, pp.222-231, 2004.
- [24] A. Nayebi, A. Shamaei, S. Meraji & H. Sarbazi Azad. An Object Oriented XML-Based Simulator, <http://www.XMulator.org>