طراحی و شبیه سازی الگوریتم های مسیریابی تحمل پذیر نقص در شبکه بر روی تراشه

نسیبه سیادتی[۱ و *] ، کریم محمدی[۲]

۱و۲- دانشگاه علم و صنعت ایران - دانشکده مهندسی برق

nasibeh.siadaty@gmail.com

(دریافت مقاله: اردیبهشت ۱۳۸۷ ، پذیرش مقاله: دی ۱۳۸۷)

چکیده

افزایش پیچیدگی طراحی مدارهای مجتمع از یک سو و نیاز به جداسازی فعالیت بخش های محاسباتی و ارتباطی در تراشه های امروزی از سویی دیگر، مسیر طراحی را به سوی سامانه های مبتنی بر شبکه روی تراشه سوق داده است.

در مقیاس های زیر میکرون تکنولوژی، تحمل پذیری نقص یک عامل با اهمیت در ارتباط با شبکه روی تراشه می شود. این مقاله الگوریتم های تحمل پذیر نقص برای استفاده در حوزه شبکه بر روی تراشه را بررسی و از لحاظ برخی پارامترهای عملکردی مقایسه می نماید. سپس الگوریتمی را در جهت برخورد با نقص های ثابت و گذرا با سطوح عملکردی مناسب، معرفی می نماید، به این نحو که در مقابل نقص های ثابتی که به طور ساختاری در تراشه موجود است بسته قابلیت انتظار و نهایتا برگشت به مبدا برای یافتن مسیری دیگر را داشته، و بیت های داده در مقابل نقص های گذرا با کدهای افزونه چرخشی و سیاست کنترل جریان سوییچ به سوییچ (سطح لینک) محافظت می شوند. با آشکار شدن نقص، بسته دور انداخته می شود چون چندین بسته از یک نوع در شبکه وجود دارد و گیرنده نیاز به درخواست ارسال مجدد ندارد چون در هر حال افزونه آن را از همان مسیر یا از مسیر دیگری دوباره دریافت خواهد کرد.

واژگان کلیدی: *تحمل پذیری نقص، نقص گذرا، نقص ثابت، شبکه بر روی تراشه، الگوریتم مسیریابی، بسته افزونه، سوییچ*

nasibeh.siadaty@gmail.com:عهده دار مکاتبات-×

۱- مقدمه

توسعه سریع در تکنولوژی سیلیکون و حرکت آن ها به سمت حوزه های نانو تکنولوژی، تراشه ها را برای جا دادن میلیاردها ترانزیستور توانا می سازد [۱]. در حین ساخت ترانزیستورها و سیم ها با اندازه ی چند ده نانومتر، عوامل زیادی باعث بروز نقص در فرآیند تولید می شوند که از جمله ی آن ها می توان به هم شنوایی[۳]، ناپایداری فرآیند[۴]، ایرادهای ناشی از تشعشعات کیهانی[۵]، مهاجرت الکتریکی[۶]، مداخله ی الکترومغناطیس[۷] وجنس مواد اشاره کرد. لذا وجود دو دسته نقص های گذرا و ماندگار در تراشه های تولیدشده امری طبیعی به نظرمی رسد.لذا طراحی سیستم هایی با قابلیت تحمل پذیری در حضور نقص های ثابت و گذرا سال هاست که مورد مطالعه می باشد. پیاده سازی الگوریتم های تحمل پذیر نقص[۸] متداول و روش های کنترل خطای موجود در حوزه شبکه بر روی تراشه[۹] به خاطر محدودیت در سطح و سربار ذخیره سازی امکان پذیر نیست بنابراین توسعه تکنیکهای مسیریابی ارتباطی جدید برای تحمل پذیری نقص ضروری است. الگوریتم سیل آسای احتمالی[۱۰] نوعی از پروتکل های سیل آسای ساده است که توسط هالبرن مطالعه شده است[۲]. این الگوریتم ها بر اساس ارتباط اتفاقی[۱۱] بوده و به هیچ نوع جداول جستجوی وسیع و یا پیکربندی مجدد نیازی ندارد و به مقدار خیلی کمی ارسال های مجدد کلی نیاز دارد. ارتباط منابع در آن ها از طریق یک طرح پخشی[۱۲]شبیه به پروتکل های خبرپراکنی تصادفی[۱۳] صورت می گیرد. الگوریتم سیل آسای جهت دار[۱۴] مانند الگوریتم های سیل آسای احتمالی می باشد با این تفاوت که احتمال انتقال بسته به هر پیوند خروجی ثابت نیست و بر اساس مقصد بسته تغییر می کند[۳].

نتایج مهم دو الگوریتم قبل در سربار انتقال خارجی و محدودکردن عملکرد کلی شبکه بود.

در الگوریتم‌گام های اتفاقی افزونه، تعدادی محدود و از پیش تعیین شده از کپی های یک پیغام داخل شبکه ارسال می شود به طوریکه این پیغام ها یک مسیر غیر مشخص را دنبال کرده و جائی که گره ای[۱۵] یکی از این پیغام ها را دریافت کند باید آن را به یکی از لینک های خروجی‌اش ارسال نماید. محاسبه مقداری منطقی برای افزون گی در این الگوریتم با استفاده از روش های مشتق شده از رشته مارکو مدل شده است. الگوریتم دیگری که یانگ در [۴] بیان کرده، مسیریابی منبع[۱۶] برای شبکه بر روی تراشه است که یک پروتکل مسیریابی ساده و کارآمد است و از دو مکانیزم کشف و نگهداری مسیر تشکیل شده، که با یکدیگر کار کرده و به گره ها اجازه کشف و نگهداری مسیریابی منبع به مقاصد دلخواه را در شبکه بر روی تراشه می دهند[۴].گره ها یا لینک هایی که دچار نقص شده اند،توسط این الگوریتم فیلتر می شوند به طوریکه مسیری که شامل گره معیوب است انتخاب نخواهد شد و انتخاب مسیرهای دیگر امکان پذیر است.

در این تحقیق با توجه به ۴ الگوریتم فوق در حوزه تحمل پذیری نقص در تراشه بر روی شبکه، الگوریتمی را ارایه خواهیم داد که راهبرد مسیریابی در آن توزیع شده[۱۷] و تطبیقی[۱۸] و مبتنی بر کانال های مجازی بوده و مکانیزم سوئیچینگ ورمهل درآن به کار رفته است. طول عمر یک بسته که در الگوریتم های قبل بر مبنای جهش های بسته تعریف می شد، در این الگوریتم با لبه بالا رونده چرخشی یک واحد کاهش می یابد و این امکان را فراهم می سازد تا بسته ها به دلیل وجود نقص های ثابت در تراشه و صفر شدن طول عمر از بین نرفته و امکان برگشت به مبدا و انتخاب مسیرهای دیگر را داشته باشند و کارایی شبکه را

3 Crosstalk
4 Process Variation
5 Cosmic Errors
6 Electro Migration
7 Electro magnetic Interference
8 Fault Tolerant
9 Network-on-Chip (NOC)
10 Probabilistic Flooding
11 Stochastic
12 Broadcast
13 Randomized gossip Protocol
14 Directed Flooding

15 Node
16 Source Routing
17 Distributed
18 Adaptive

جدول۱ - طبقه بندی الگوریتم ها بر اساس پارامترهای قابل پیکربندی

پارامترها / الگوریتم ها	مکانیزم سوئیچینگ		فرمت بسته		راهبرد مسیریابی				تعداد بسته های افزونه تولیدی سوییچ منبع			طول عمر		انتخاب پورت خروجی		کدهای کنترل نقص
									دینامیک							
	بسته	ورمهل	پیکرچه	تفکیک شده	تطبیقی	انتفاقی	توزیع شده	منبع	احتمالی	روابط ریاضی	استاتیک	clk	s-s	با الویت	بدون الویت	
سیل آسای احتمالی	✓	-	✓	-	-	✓	✓	-	✓	-	-	-	✓	-	✓	-
سیل آسای جهت دار	✓	-	✓	-	-	✓	✓	-	✓	✓	-	-	✓	✓	-	-
N-Random Walk	✓	-	✓	-	-	✓	✓	-	-	✓	-	-	✓	-	✓	-
شبکه فوق العاده اتفاقی	✓	-	✓	-	-	-	-	✓	-	-	-	-	✓	✓	-	-
پیشنهادی	-	✓	-	✓	✓	-	✓	-	-	-	✓	✓	-	✓	-	✓

بالا می برد. همچنین وجود سیاست کنترل خطا که در هر سوییچ و با کدهای افزونه چرخشی[19] اجرا می‌شود[۵]، امکان آشکارسازی تغییرات در جریان بیت داده را که به علت نقص های گذرا صورت می پذیرد، فراهم ساخته و سپس نسبت به از بین بردن بسته اقدام می‌شود. از آن جایی که چندین بسته از یک نوع درشبکه وجود دارد، گیرنده نیاز به درخواست ارسال مجدد ندارد چون افزونه آن را از همان مسیر یا از مسیر دیگری دوباره دریافت خواهد کرد. طبقه بندی الگوریتم ها بر اساس پارامترهای قابل پیکربندی آن ها در جدول ۱ نشان داده شده است.

۲- الگوریتم پیشنهادی (XYFT)[20]

۲-۱ راهکار مسیریابی پیشنهاد شده

در یک شبکه بر روی تراشه سوییچ ها مسئول انجام مسیریابی بین گره ها می‌باشند. هر سوییچ شامل یک مسیریاب برای ایجاد یک مسیر از درگاه ورودی به خروجی تعدادی بافر برای نگهداری اطلاعات میانی و یک واحد داوری برای تعیین الویت ؛برای هنگامی که چندین درخواست به طور هم زمان به یک سوییچ می رسند، می باشد. یکی از مهم ترین جنبه های پیاده‌سازی یک سوییچ، الگوریتم مسیریابی است.

مسیریابی می تواند در گره مبدا و یا به صورت توزیع شده انجام گیرد و از منظر دیگر تطبیقی و یا قطعی باشد. در این الگوریتم، راهبرد مسیریابی توزیع شده و تطبیقی در نظر گرفته شده تا با استفاده از وضعیت شبکه، مسیر بین مبدا و مقصد برای جلوگیری از تراکم[21] و دور زدن قسمت‌های معیوب انتخاب گردد. و سوییچ تابعی از مسیریاب و انتخاب گر خواهد شد.

با توجه به مکانیزم سوییچینگ ورمهل به کار گرفته شده در این الگوریتم، بسته ها به واحدهای فلیت[22] تقسیم شده‌اند. تنها فلیت سرآیند اطلاعات مسیریابی را دارد و بقیه ی فلیت هایی که بسته را می سازند باید همان مسیری را که برای سرآیند ذخیره شده است دنبال کنند. در فلیت اول، آدرس سوییچ مقصد و در فلیت دوم، تعداد فلیت‌هایی که در بار مفید بسته می‌باشد، قرار داده مـــــی شود. هرسوییچ در شبکه یک آدرس منحصر به فرد دارد، و برای سادگی عمل مسیریابی روی شبکه، موقعیت در مختصات **XY** بیان می‌شود، که **X** موقعیت افقی و**Y**موقعیت عمودی را بیان می کند. بدین صورت که اگر منبعی بسته‌ای را برای ارسال داشته باشد آن

19 Cyclic Redundancy codes (CRC)
20 XY Fault Tolerant

21 Congestion
22 Flow Control Unit

را به واحد سوییچ خود تحویل می دهد. سوییچ، **R** بسته افزونه از آن بسته تهیه کرده و به سوییچ های مجاور می فرستد که **R** برای تمام بسته ها ثابت بوده و بر اساس میزان تحمل پذیری نقص لازم، تخمین زده می شود. در این الگوریتم نیز مانند الگوریتم های قبل، برای هر بسته یک طول عمر در نظر گرفته می شود ولی این طول عمر در هر سوییچ، تنها یک واحد کاهش پیدا نمی کند، بلکه با هر لبه ی کلاک یک واحد کاهش پیدا می کند. این مطلب قابلیت انتظار را برای بسته در صورت عدم امکان عبور از پورت خروجی مورد نظر یا پورتهای مفید دیگر به وجود می آورد. یعنی بسته می تواند حداکثر **L** کلاک در بافر سوییچ منتظر بماند یا به عبارت دیگر، طول عمر در هر سوییچ می تواند حداکثر **L** کلاک کاهش پیدا کند. اگر در طول این مدت پورت مورد نظر آزاد گردد بسته منتقل می شود، اگر طول عمر صفر شود بسته دور ریخته می شود؛ قابلیت انتظار باعث می شود که تعداد بسته های افزونه کاهش پیدا کند و در نتیجه از بار شبکه کاسته شده و کارآیی شبکه بالا می رود. (این از مزایای روش پیشنهادی می باشد.) در غیر این صورت سوییچ فرض می کند که سوییچ های مجاورش یا از کار افتاده اند یا پیوند بین آنها قطع شده است. یعنی یک نقص ثابت در شبکه موجود است، بنابراین بسته را برگشت می دهد تا مسیر دیگری را بپیماید. اگر در حین پیمایش بسته میان سوییچ ها نقص گذرایی هر گونه تغییری در جریان بیت بسته به وجود آورد متد **CRC** که در هدر بسته می باشد قابلیت آشکارسازی تغییرات را داشته و بسته ای که با این گونه نقص مواجه شده است دور ریخته می شود و به این ترتیب راهکاری برای پوشش نقص در نظر گرفته شده تا این نقص منجر به ایجاد خطا در سیستم نگردد.

وقتی سوییچی بسته ای را دریافت کند آدرس خودش را با آدرس سوییچ مقصد بسته که در فلیت سرآیند ذخیره شده است با هم مقایسه و مطابق الگوریتم **XY** عمل می نماید. اگر آدرس منبعش را به عنوان مقصد داشته باشد بسته را به منبعش تحویل می دهد وگرنه آن را به یکی از کانال های مجازی پورت خروجی می فرستد که البته ابتدا سعی می کند در جهت مقصد باشد ولی اگر پورت های مفید اشغال باشند همان گونه که در بالا ذکر گردید، به اندازه مدت زمان انتظار، منتظر می ماند. اگر چندین بسته از پورت های ورودی مختلف، به طور هم زمان قصد عبور از یک کانال خروجی را داشته باشند قضاوت چرخشی[۲۳] مرکزی الویت دسترسی را به بسته های ورودی اهدا می دهد. اولویت کانال های مجازی پورت خروجی، تابعی از آخرین پورتی است که باید یک درخواست مسیریابی را پذیرش کند. به عنوان مثال اگر پورت ورودی محلی آخرین پذیرش درخواست مسیریابی را داشته باشد اولویت پورت ها به ترتیب شرق، غرب، شمال، جنوب و محلی خواهند بود. این روش ضمانت می کند که کلیه نیازهای ورودی برآورده خواهند شد و قحطی زدگی[۲۴] به وجود نخواهد آمد. زمانی که الگوریتم مسیریابی **XY** پورت خروجی مناسب و آزاد را پیدا کند، اتصال بین کانال مجازی پورت ورودی و خروجی به وجود می آید و بردارهای سوییچینگ شامل ورودی، خروجی و آزاد در جدول سوییچینگ بروز خواهند شد. بردار آزاد مسئول تغییر حالت کانال مجازی خروجی از وضعیت آزاد (۱) به مشغول (۰) می باشد. بردار ورودی کانال مجازی ورودی را به کانال مجازی خروجی و خروجی کانال مجازی، خروجی را به یک کانال ورودی متصل می نماید. برخی از مزایای الگوریتم پیشنهادی در بخش های ۲-۲ و ۲-۳ بیان می شود.

[23] **Round-robin**
[24] **Starvation**

جدول۲: ویژگیهای شبکه بر روی تراشه در محیط شبیه سازی

مشخصات	توصیف
توپولوژی / ابعاد	مش دوبعدی / ۵×۵
مسیریابی	**X-Y** تطبیقی
سوئیچینگ	ورمهل
کنترل جریان	سنجش اعتبار
قضاوت	قضاوت چرخشی
تخصیص حافظه	پورت و کانال مجازی
سایز بسته/ تعداد کانالهای مجازی	۶۴ بیت/۲
سایز بافر	۸ بیت
توزیع تولید بسته	پوآسون
الگوی ترافیک	یکنواخت

۲-۲ طول عمر

در الگوریتم پیشنهادی، برای هر بسته یک طول عمر در نظر گرفته می شود. طول عمر، در هر سوییچ با هر لبه کلاک یک واحد کاهش پیدا می کند و در صورتی که صفر شود بسته دور ریخته می‌شود. طول عمر باعث کاهش تعداد بسته های موجود در شبکه می‌شود و از اشباع در شبکه جلوگیری می کند. طول عمرهای خیلی کوچک، کارآیی خیلی کمی را به وجود می‌آورند. این لزوما به خاطر نقص ها نیست بلکه به خاطر دور ریختن بسته ها می‌باشد. واضح است طول عمر بالا نیز باعث تولید تعداد زیادی بسته در شبکه برای مدت طولانی می‌شود و در نتیجه تراکم در شبکه را به وجود می‌آورد. بنابراین طول عمر باید تا حدی کوچک انتخاب شود تا هم تحمل پذیری بالایی به وجود آید و هم از تراکم در شبکه جلوگیری شود.

۲-۳ میزان انتظار

قابلیت انتظار باعث می‌شود که تعداد بسته‌های افزونه کاهش پیدا کند چون دیگر بسته ها به علت پر بودن بافر دور ریخته نمی‌شوند. در نتیجه از بار شبکه کاسته می‌شود و کارآیی شبکه بالا می‌رود. انتظار خیلی کم، کارآیی خیلی کمی را به وجود می‌آورد که این به علت برگشت دادن بسته‌ها می‌باشد. واضح است انتظار بالا نیز باعث کاهش کارآیی خواهد شد چون بسته ها به خاطر صفر شدن طول عمر دور ریخته می‌شوند. بنابراین میزان انتظار باید به اندازه‌ای باشد تا هم از برگشت زیاد بسته‌ها و هم از دور ریختن بسته‌ها جلوگیری شود.

۳- ارزیابی نتایج شبیه سازی

به منظور ارزیابی کارایی الگوریتم ها، این الگوریتم در محیط شبیه سازی **Visual Studio 6** و با زبان شئ گرای ++ **C** برای نوشتن توابع و کلاسهای مورد نظر کدنویسی شده و با الگوریتم‌های موجود مقایسه می‌گردد. نتایج استخراج شده در مدل ما، از شبکه مش دو بعدی با ابعاد ۵×۵ بررسی شده و برای هر کانال فیزیکی ورودی و خروجی ۲ کانال مجازی در نظر گرفته شده است. زمانی که **n** کانال مجازی استفاده شود بافری به عمق **d/n** خواهیم داشت (که **d** سایز فلیتها است) و به هر یک از کانال های مجازی مرتبط می شود[۷]. در هر شبیه سازی کلیه منابع سالم با استفاده از توزیع پوآسون و الگوی ترافیکی یکنواخت، بسته هایی را به مقاصد مورد نظر ارسال می کنند. ویژگی های شبکه بر روی تراشه در محیط شبیه سازی در جدول ۲ نشان داده شده است.

جدول۳: مقایسه الگوریتم ها بر اساس معیارهای ارائه شده

طول عمر	سربار شبکه	تاخیر	سطح تحمل پذیری نقص		نرخ تولید بسته	الگوریتم
			نقص های گذرا	نقص های دائمی		
متوسط	خیلی بالا	بالا	متوسط	متوسط	۰،۰۲	سیل آسای احتمالی
متوسط	بالا	متوسط	بالا	بالا	۰،۰۲	سیل آسای جهت دار
متوسط	بالا	بالا	متوسط	متوسط	۰،۰۵	**Random Walk**
متوسط	متوسط	متوسط	متوسط	خیلی بالا	۰،۰۵	شبکه فوق العاده اتفاقی
بالا	متوسط	کم	بالا	بالا	۰،۰۶	پیشنهادی
متوسط : ۷۰>طول عمر>۶۰ بالا : طول عمر>۷۰	متوسط : ۴۰> سربار بالا : ۶۰>سربار>۵۰ خیلی بالا: سربار >۶۰	کم : ۳> تاخیر متوسط : ۳،۵> تاخیر >۳ بالا : تاخیر >۴ کلاک	متوسط : ۸۰> تحمل پذیری >۷۰ بالا : تحمل پذیری >۸۰	متوسط : ۸۰> تحمل پذیری >۶۰ بالا : ۹۰> تحمل پذیری >۸۰ خیلی بالا: تحمل پذیری		معیار عبارات کیفی

۴. بررسی معیارهای ارزیابی

برای ارزیابی کارایی الگوریتم های تحمل پذیر نقص، چهار معیار در نظر گرفته می شود: طول عمر، تحمل پذیری نقص (نقص های دائمی، نقص های گذرا)، سربار شبکه و تاخیر. مقایسه معیارهای ذکر شده در بخش های ۴-۱، ۴-۲، ۴-۳ و۴-۴ آورده شده است و به صورت تفکیکی در جدول ۳ نشان داده شده است. در نمودارهای بخش های نتایج، **P** معرف احتمال ارسال بسته در الگوریتم های سیل آسای احتمالی و جهت دار، **N** معرف تعداد افزونه گی های بسته در الگوریتم گام های تصادفی و **R** نشان دهنده تعداد کپی باشد.

۴-۱ طول عمر

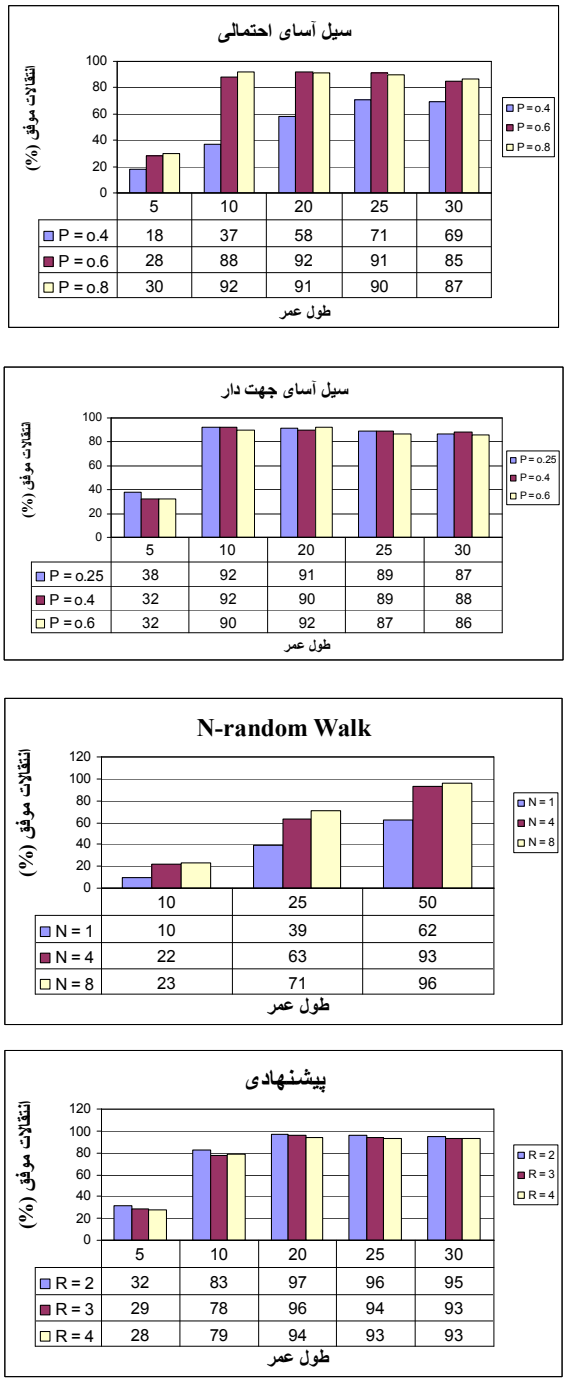
در الگوریتم های موجود و پیشنهادی، برای هر بسته یک طول عمر در نظر گرفته می شود. طول عمر در هر سوییچ و یا با هر لبه کلاک یک واحد کاهش می یابد و در صورتی که صفر شود، بسته دور ریخته می شود. طول عمرکوچک کارایی خیلی کمی را به دست می دهد. این لزوما به خاطر نقص نیست، بلکه به خاطر دور ریختن بسته ها می باشد. واضح است طول عمر بالا نیز باعث تولید تعداد زیادی بسته در شبکه برای مدت طولانی می شود و در نتیجه تراکم شبکه را افزایش می دهد. بنابراین طول عمر باید تا حدی کوچک انتخاب شود تا هم تحمل پذیری نقص در سطح بالایی به وجود آید و هم از تراکم در شبکه جلوگیری شود. نتایج شبیه سازی طول عمر در شکل۱، تاثیر انتخاب مقادیر مختلف طول عمر را برای الگوریتم های مختلف نشان می دهد.

در شبیه سازی ها، طول عمر در الگوریتم های سیل آسا ۱۰ سوییچ و در الگوریتم پیشنهادی ۲۰ کلاک انتخاب شده است.

نقص های دائمی همیشه به دلیل یک عیب فیزیکی به وجود می‌آیند که می توانند مستقیما بعد از ساخت دیده شوند یا هنگام استفاده از تراشه رخ دهند. به عبارت دیگر آنها تحت هر شرایطی می توانند به وجود آیند و ممکن است یک یا چند خط را تحت تاثیر قرار دهند. نقص های دائمی متفاوت، می توانند اثرات متفاوتی نیز داشته باشند.

نقص‌های دائمی شامل گره هایی می شود که در دریافت و ارسال بسته ها ناتوان هستند. این نقص ها درصدی از کل اجزای شبکه می باشند که کاملا از کار افتاده اند. شکل ۲ تحمل پذیری نقص الگوریتم‌های مختلف را برای نقص های دائمی نشان می دهد. همان طور که دیده می‌شود در سطوح پایین نقص های ثابت، الگوریتم گام های تصادفی تقریبا خوب عمل می‌کند اما برای درصد ۱۰ به بالا الگوریتم های سیل آساياحتمالی و جهت دار بهتر می‌باشند. تحمل پذیری نقص گذرا در الگوریتم های شبکه فوق العاده اتفاقی و سیل آسای جهت دار سطوح مشابهی دارد در حالی که نقص در الگوریتم سیل آسای احتمالی اندکی کمتر از دو الگوریتم دیگر است.

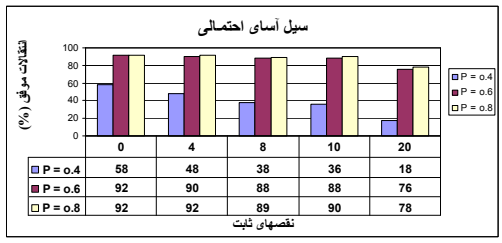
در نقص های ثابت، الگوریتم شبکه فوق العاده اتفاقی بهترین درصد انتقالات موفق را دارد. دلیل این است که الگوریتم مذکور مکانیزمی برای گره های فالتی ثابت که در طول فرآیند کشف مسیر انجام شده است، داشته و پس از کشف مسیرهای معیوب، آنها را از کش خود خارج می‌کند. در الگوریتم پیشنهادی با نسبت دادن طول عمر بسته ها به لبه کلاک قابلیت انتظار برای آنها در بافر سوییچ ها به وجود آمده و در صورت عدم عبور بسته، آن را به فرستنده بر می گرداند لذا از سطح بالایی از تحمل پذیری نقص برخوردار است.

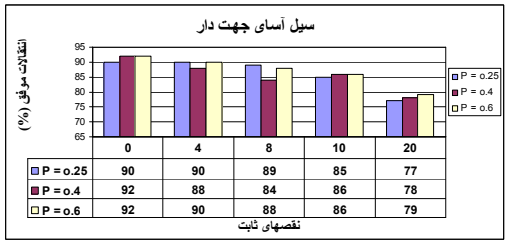


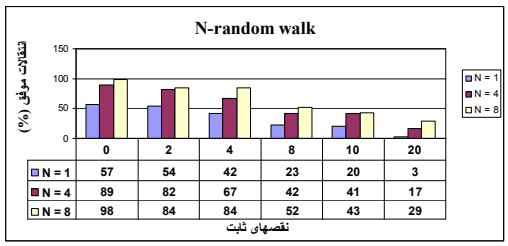
شکل ۱- بررسی طول عمر بر روی انتقالات موفق در حضور ۲٪ نقص ثابت و ۱٪ نقص گذرا

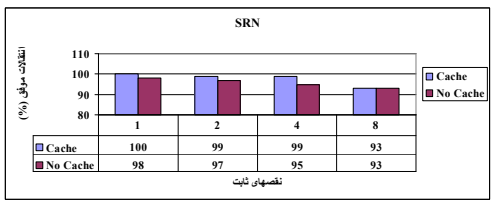
۴-۲ تحمل پذیری نقص

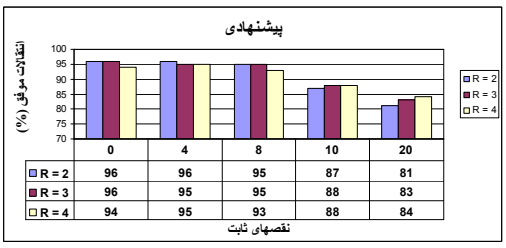
تحمل پذیری نقص، قابلیت اطمینان هر الگوریتم را در مسیریابی بسته‌ها با وجود نقص‌های دائمی و گذرا، ارزیابی می کند[۶].





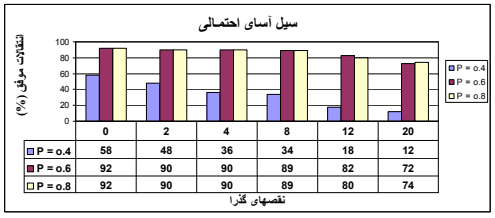


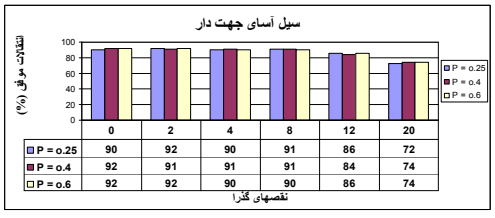


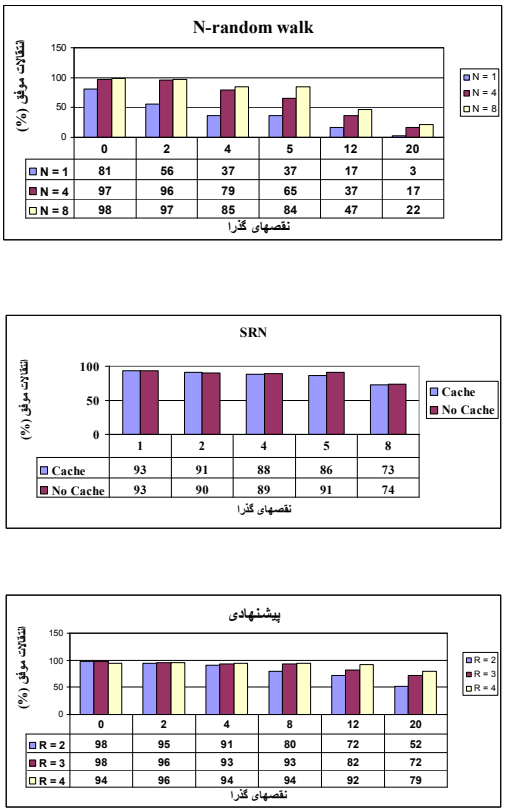


شکل ۲- اثر نقص های دائمی روی انتقالات موفق با حضور ۲٪ نقص گذرا

در مقابل نقص های دائمی، نقص‌های گذرا وجود دارند که تنها برای یک مدت زمان معین می باشند. هر چند ممکن است به صورت مستمر آشکار و ناپدید شوند. (انتظار می رود در تکنولوژی های آینده نقص‌های گذرا غالب شوند.) نقص های گذرا آن هایی هستند که با درصد شانسی، بسته می تواند دریافت و ارسال شود. این نقص ها شامل سرریز بافر و همچنین آنهایی که داده بسته را تغییر می دهند می باشد. در شبیه سازی، این نقص ها به صورت تصادفی به سیستم تزریق شده و برخی بیت های بسته را تغییر می‌دهد و از آن جا که نرخ های تزریق نقص پایین انتخاب شده است، استفاده از کدهای تشخیص خطا که سربار سطحی کمتری نسبت به کدهای تصحیح خطا دارند، استفاده شده است. چون افزونه بسته از مسیری دیگر و یا همان مسیر دریافت می گردد در این شبیه سازی برای آشکارسازی تغییرات از کدهای چرخشی که قدرت آشکارسازی بالایی دارند برای محافظت از بسته ها استفاده شده که پس از آشکار سازی بسته دور ریخته خواهد شد.. شکل ۳ تحمل پذیری نقص الگوریتم های مختلف را برای نقص های گذرا نشان می دهد. همان طور که دیده می شود، الگوریتم های سیل آسای احتمالی و **Rw** تقریبا در یک سطح و الگوریتم های سیل آسای جهت دار و شبکه فوق العاده اتفاقی نیز در سطح بالاتری می باشند. نرخ های نقص بیش از ۸٪ توان عملیاتی الگوریتم ها را کاهش می‌دهد البته این تعداد نقص در عمل خیلی بالا می باشد ولی در عین حال می توان در الگوریتم ها با افزایش کیفیت بسته های افزونه و قابلیت تصحیح خطا درصد انتقالات موفق را بالا برد.





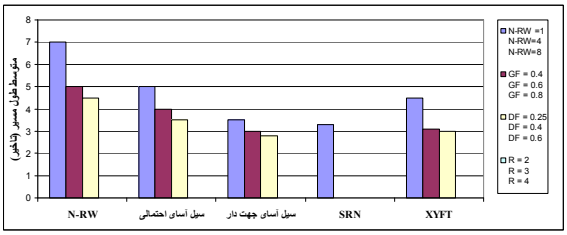


شکل۳- اثر نقص های گذرا روی انتقالات موفق (نرخ نقص دائمی=۰)

۴-۳ تأخیر

تأخیر مدت زمانی است که طول می کشد تا بسته ای از مبدا به مقصد برسد. و این با شمارش تعداد جهش‌هایی که بسته داشته تا به مقصد برسد صورت می گیرد. نظر به این که در الگوریتم ها از بسته های افزونه استفاده شده است، تنها زمان دریافت اولین بسته به عنوان تاخیر آن بسته در نظر گرفته می‌شود. در شکل۴ تاخیر در چهار الگوریتم نشان داده شده است. همان طور که مشخص است الگوریتم های **N-Rw** و سیل آسای احتمالی بیشترین تاخیر را دارند. الگوریتم های شبکه فوق العاده اتفاقی و سیل آسای جهت دار تمایل به کوتاهترین مسیر دارند لذا تاخیر کمتری دارند. تاخیر در شبکه فوق العاده اتفاقی به علت مکانیزم کش و زمان تنظیمات اولیه می باشد. در الگوریتم سیل آسای جهت دار، افزونه گی اثر کمی روی تاخیر دارد که این امر به خاطر تمایل الگوریتم برای پیدا کردن جهت مقصد می باشد.

در الگوریتم پیشنهادی نیز با توجه به تعداد مشخص و محدودی از بسته های افزونه و با توجه به راهکار مسیریابی کمترین مقدار تاخیر را داریم.



شکل۴- مقایسه تاخیر الگوریتم ها در حضور ۲٪ نقص ثابت و ۱٪ نقص گذرا

۴-۴ سربار شبکه

برای بررسی میزان تراکم شبکه به صورت کیفی، از آن تحت عنوان سربار شبکه استفاده نموده ایم. سربار شبکه با در نظر گرفتن کل بسته های فرستاده شده در شبکه محاسبه می شود. تعداد کل بسته هایی که از مسیر مبدا به مقصد تولید می شوند، برابر است با تعداد بسته‌هایی که سوییچ مبدا تولید می کند بعلاوه مجموع تعداد بسته‌هایی که توسط سوییچ های مسیریاب در طول مسیر تولید می شوند.

با این تعریف در الگوریتم سیل‌آسای احتمالی در بدترین مورد اگر فرض شود سوییچ مبدا بسته را به هر چهار پورت خروجی اش ارسال کرده و پورت ها نیز آزاد باشند، پس چهار سوییچ مجاور هر کدام یک بسته دارند که آن ها نیز به هر چهار پورت خروجی خود ارسال خواهند کرد. اگر فرض شود فاصله سوییچ مبدا تا مقصد **R** باشد و در این فاصله **n** سوییچ برای مسیریابی وجود داشته باشند **4n** بسته تنها در مسیر منتهی بین مبدا و مقصد تولید خواهد شد که از میان این بسته‌ها تنها یک بسته مفید بوده و سه بسته دیگر افزونه آن بوده و جز سربار محسوب می شوند پس سربار شبکه بالا خواهد بود.

در الگوریتم سیل‌آسای جهت‌دار با توجه به اینکه فاصله سوییچ مبدا و سوییچ‌های مجاورش تا مقصد محاسبه و سپس احتمال ها مشخص می‌شوند، لذا بسته های سربار کمتری در طول مسیر، نسبت به الگوریتم سیل آسای احتمالی تولید می‌شود. در الگوریتم پیشنهادی با حالت بیشترین انتخاب در تعداد بسته افزونه، (یعنی چهار) که تنها در سوییچ مبدا تولید می‌گردد، کمترین سربار را خواهیم داشت.

۵. نتیجه گیری

معمولا شبکه‌های داده مثل پروتکل اینترنت یا **ATM** الگوریتم‌های پیچیده‌ای برای تحمل پذیری نقص دارند. که هیچ کدام از آن ها در حالت عادی شان به علت پیچیدگی بیش از حدشان برای شبکه بر روی تراشه ها مناسب نبوده، و به منابعی نیاز دارند که روی تراشه قابل دسترسی نیستند[۸]. لذا بررسی الگوریتم‌های دینامیک و مکانیزم های ارسال مجدد در حوزه این تحقیق نمی‌باشد. هر چند در الگوریتم های مورد بحث تدابیری برای تحمل پذیری نقص در نظر گرفته شده است. یعنی جداول مسیریابی ثابت و از پیش تعیین شده‌ای وجود نداشته و همان طور که اشاره گردید جداول سوییچینگ نسبت به وضعیت فعلی پورتها کمی شده و سپس نسبت به ارسال بسته تصمیم گیری می‌شود.

الگوریتم‌های موجود سیل آسای احتمالی، سیل آسای جهت‌دار، **N-RW** و شبکه فوق العاده اتفاقی می باشند. با توجه به نتایج جدول ۳، الگوریتم‌های سیل آسا ساده و سبک هستند و تحمل پذیری نقص در آنها بر مبنای ارسالهای زیاد از یک بسته می‌باشد اما باعث سربار ارتباطی قابل توجهی می باشند. لذا پیاده سازی این گونه الگوریتم‌ها به نرخ تولید بسته کم محدود خواهد شد. الگوریتم سیل آسای جهت‌دار با نرخ ارسال داده کمتر، باعث کاهش در سربار شبکه می‌شود.

الگوریتم **N-RW** با بهبودهایی از قبیل تنظیم دینامیکی بسته‌های افزونه و انتخاب مسیرهای منحصر به فرد و غیر یکنواخت تحمل پذیری نقص را با نرخ تولید بسته کمتری انجام می دهد. اما در نرخ های نقص فراتر از ده درصد، توان عملیاتی آن از بین می‌رود. نهایتا الگوریتم شبکه فوق العاده اتفاقی با هدف کمینه کردن سربار مسیریابی و به دست آوردن تحمل‌پذیری نقص ارائه شده است. تحمل پذیری شبکه فوق العاده اتفاقی در برابر نقص‌های گذرا همانند الگوریتم های سیل آساست، افزون بر این که با تعداد محدود پخش بسته‌های پیام در مکانیزم کشف مسیر و مکانیزم تک پراکنی در انتقال داده کارایی بهتر و تحمل پذیری نقص ثابت بالاتری دارد. همچنین بهبودها در آن با توسعه مسیریابی دینامیک با کش پیام های درخواست مسیر، برای کنترل شرایط جاری شبکه به کار می رود و در نهایت در الگوریتم پیشنهادی با تولید بسته های افزونه به مقدار ثابت و از پیش تعیین شده سربار شبکه کمتر خواهد بود. انتظار بسته‌ها در سوییچ ها برای مدت زمان معین باعث افزایش کارایی و دور ریخته نشدن بسته ها با وجود نقص های دائمی است. به کارگیری کدهای تشخیص خطا نیز حفاظت از بسته ها در برابر نقصهای گذرا است که در هر دو حالت سطح بالایی از تحمل پذیری نقص حاصل می‌شود در حالی که سربار شبکه نیز در سطح متوسط است. از آنجا که طول عمر بسته‌ها به روش بهینه تری تعریف شده، لذا در این خصوص درصد بالاتری از انتقالات موفق را خواهیم داشت.

مراجع

[1] Semiconductor Industry Association, International Technology Roadmap for Semiconductors, World Semiconductor Council, Edition 2005, 2005.

[2] L. Li, J. Halpern, and Z. Haas, " Gossip based ad hoc routing, In Networking "IEEE/ACM Transactions on June 2006, Volume 14, Issue 3, pp. 479-491.

[3] M.Pirretti, G.M.Link, Brooks, R.R, Vijaykrishnan, N, Kandemir, M, Irwin, M.J., "Fault tolerant algorithms for network-on-chip interconnect," In Proceedings IEEE Computer society Annual Symposium on 19-20 Feb. 2004, pp. 46 – 51.

[4] Kim, Young Bok; Kim, Yong-Bini " Fault

Tolerant Source Routing for Network-on-chip," Dept. of Electrical and Computer Engineering ,Northeastern University, Boston, MA ,USA, DFT apos;07. 22nd IEEE International Symposium on Volume , Issue , 26-28 Sept. 2007 Page(s):12 - 20.

[5] Daniele Rossi, Paolo Angelini, Cecilia Metra " Configurable Error Control Scheme for NOC Signal Integrity," D.E.I.S. University of Bologna Viale Risorgimento 2, 40136 Bologna, Italy ,2007.

[6] Barry W.johnson., "Design and analysis of fault-tolerant digital systems," . Addison-Wesley,1989.

[7] A. Mello, L. Tedesco, " Virtual Channels in Networks on Chip: Implementation and Evaluation on Hermes NoC," IFIP Very Large Scale Integration (VLSI-SOC), (FACIN-PUCRS) SBCCI'05, September 4-7, 2005.

[8] Muhammad Ali, Michael Welzl, Sven Hessler "A Fault tolerant mechanism for handling Permanent and Transient Failures in a Network on Chip," Institute of Computer Science, University of Innsbruck, Austria Sybille Hellebrand Institute of Electrical Engineering, University of Paderborn, Germany. International Conference on Information Technology (ITNG'07)2007.