تصميم نموذج أولي وتنفيذه لنظام تكييف الدفق عبر شبكة حاسوبية اعتماداً على هوية التطبيقات

المهندس أسعد جاموس*

د. محمد نوار العوا ** **د. ماهر سليمان*****

الملخص

يتناول هذا البحث دراسة نظرية وعملية لمنظومة تكييف التدفقات في الشبكة بحسب التطبيقات، وذلك من أجل المساعدة في التحكم وإدارة موارد الشبكة التي منها سعة الوصلة في الشبكة، حيث قُسِّمَتِ السعة الكلية بين التطبيقات بشكل فعال حسب الأهمية والأولوية، دُرِسَتْ في هذا البحث دراسة طرائق التصنيف المختلفة للرزم، ومحدودية تلك الطرائق وآلية تحديد هويات التطبيقات وطرائق نمذجتها والتعابير النظامية بوصفها إحدى الطرائق لتمثيل الهويات، وكذلك آليات تكييف الدفق، ومن ثم تصميم نموذج عام لنظام برمجي يقوم بالبحث عن هوية التطبيق ضمن الرزم العابرة للمسيّر Router وتصنيف تلك الرزم بحسب تلك الهويات، ومن ثم تطبيق أنظمة جودة الخدمة عليها. نُفِّذَتْ مكونات المنظومة باستخدام الوحدات البرمجية والبرامج المفتوحة المصدر الموجودة في بيئة نظام التشغيل Linux. بعد ذلك جرى تنفيذ بعض الاختبارات التحليلية لقياس أثر البحث عن الهوية ومطابقة التعابير النظامية ضمن أعماق الرزم في الأداء، ومناقشة قيود المنظومة ومحدودياتها.

الكلمات المفتاحية: تكييف الدفق، تصنيف الرزم، التعابير النظامية، هوية التطبيق، كفاءة الخدمة.

[1] * أعد البحث في سياق رسالة الماجستير للمهندس أسعد جاموس بإشراف الدكتور المهندس محمد نوار العوا والدكتور المهندس ماهر سليمان

** قسم النظم والشبكات الحاسوبية - كلية الهندسة المعلوماتية، جامعة دمشق

*** المعهد العالي للعلوم التطبيقية والتكنولوجيا - دمشق

1 - مقدمة:

مع الانتشار الكبير للتطبيقات الشبكية والاعتماد عليها في المجالات الحكومية والاقتصادية والخدمية كلّها وظهور كثير من التطبيقات الشرهة للسعات مثل برامج الند للند P2P ومشاركة الملفات والفيديو ولإدارة وصلة الشبكة بشكل فعال حسب الأهمية والأولوية كانت الطرائق التقليدية لبناء أنظمة جودة الخدمة تعتمد فقط على معلومات التحكم المتوافرة في ترويسات الرزم مثل رقم المنفذ المصدر أو الوجهة مثلاً الرزم التي قيمة المنفذ فيها 80 تتبع لتطبيقات الوب وهذه الطريقة لم تعد فعالة في الوقت الراهن لأن كثيراً من التطبيقات الحالية تستخدم قيماً اعتباطية لأرقام المنفذ وكذلك أصبح شائعاً تحميل بيانات تطبيق باستخدام تطبيق آخر على سبيل المثال إن البرتوكول HTTP يستخدم عادة لاستعراض صفحات الويب WWW ولكن كثيراً من التّطبيقات الحالية كبرامج الند للند أصبحت تستخدم البرتوكول نفسه لتشغيل تطبيقاتها وباستخدام رقم المنفذ نفسه، ومن ثمَّ إذا اعتمد المصنف (classifier) فقط على الحقل "رقم المنفذ" سوف يصنف هذا التّطبيق على أنه استعراض ويب في حين بالحقيقة هو برنامج ند للند. لذلك أصبح ضرورياً إيجاد الآليات والتقنيات التي تستطيع كشف تلك التطبيقات ومن تلك الآليات تحديد بصمات هويات - للتطبيقات المختلفة والبحث عنها ضمن أعماق الرزم.

2 -الدراسة المرجعية:

تعتمد طريقة البحث على تقديم دراسة نظرية تتضمن مفهوم تصنيف الرزم ضمن طبقات TCP/IP ومعايير التصنيف وميزات ومحدودية الاعتماد على معلومات ترويسات الطبقات خلال عملية التصنيف ومن ثم دراسة مفهوم هوية التطبيق وطرائق التعبير عن تلك الهوية والمحددات الأساسية لنظام البحث عن هوية الرزم المارة عبر المسيّر وكذلك دُرِسَتِ التعابير النظامية كونها إحدى الطرائق التي توّصف هويات التطبيقات وآلية عملها وتعقيدها الزمني بعد ذلك دُرِسَتْ تقنيات تكييف الدفق المختلفة للتدفقات وآلية عملها وكيف يمكن مكاملتها مع وحدة التصنيف.

بعد ذلك صُمِّمَ النظام الأولي للمنظومة ونُفِّذَ على حاسوب يعمل ضمن بيئة نظام التشغيل Linux ويعمل كمسيّير ضمن الشبكة وأخيراً أُجْرِيَتِ الدراسة التحليلية وتتضمن اختبارات لقياس أثر البحث عن الهوية في أعماق الرزم ونوقِشَتْ قيود المنظومة.

2 -1 تصنيف الرزم:

التصنيف هو تقسيم الرزم إلى فئات متعددة بناءً على وجود صفات محددة بحيث توضع الرزم المتشابهة جميعها في فئةٍ واحدة إن عمليّة تصنيف الرزم الّتي تمر عبر مسيّر وفق قواعد محددة إلى مجموعات مختلفة تدفقات flows تدعى packet classification نحتاج إلى تصنيف الرزم من أجل عمل تطبيقات جودة الخدمة QoS يُجرى التصنيف في طبقة الشبكة اعتماداً على العناوين الرقمية (IP) أمَّا التصنيف في طبقة النقل فيتم اعتماداً على أرقام المنفذ حيث يرتبط كل تطبيق بقيمة معنية وهي عملية غير مكلفة كون الوصول إلى تلك القيم هي عملية مباشرة لكن الاعتماد على تلك الحقول للتصنيف غير كافٍ لأنَّ التطبيقات أصبحت تستخدم أرقام منافذ عشوائية أو تحمل بياناتها باستخدام تطبيق آخر على سبيل المثال كثير من تطبيقات الند للند تستخدم البروتوكول HTTP في عمليات النقل.

2 -2 هوية التطبيق:

لكلّ تطبيق طريقة تخاطب خاصة به يستخدمها خلال الاتصال والرسائل الّتي يتم تبادلها وإذا استطعنا معرفة طريقة التخاطب الخاصة بالتّطبيق هذا أو تحديد أثر (بصمة) للتّطبيق ضمن بيانات الرزم فمن الممكن مراقبة الرزم واكتشاف التّطبيق، ومن ثمَّ يمكن تصنيف الرزم على مستوى التّطبيق بحدّ ذاته البحث عن تلك الهويّة ضمن الرزم يسمى بالتصنيف اعتماداً على الهوية (signature based packet classifier).

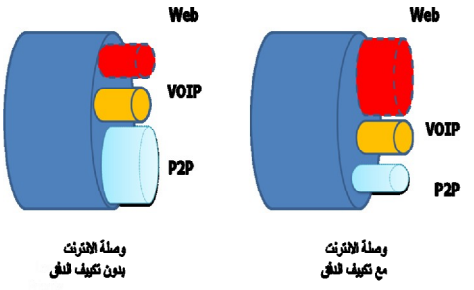
بعد تحديد الهويّة واكتشاف بصمة التّطبيق كيف يتم تمثيل تلك الهوية وقابلية برمجتها على تجهيزات متنوعة قد تكون برمجية أو مادية ولاسيّما أن الهوية قد تكون معقّدة فالهوية قد تكون عبارة عن مجموعة من المحارف المحدودة والمتتالية القابلة للطباعة وقد تحوي مجموعة محارف تحكم (غير قابلة للطباعة Binary) وقد تكون غير متتالية وموزّعة على امتداد الرزمة أو مجموعة من الرزم. وإحدى الطرائق لنمذجة الهويات هي باستخدام التعابير النظامية[1].

التعابير النظامية: هي أسلوب لوصف مجموعة من المحارف وتعرّفها ضمن نص عن طريق وصف مكوناتها من رموز، ووصف علاقات تلك الرموز من توال وتكرار، وذلك بكيفية يمكن لخوارزمية أن تفسرها وتطبقها على نص مُعطى لاستخراج الجزء الذي يطابق التعبير النظامي.

تكمن أهمية التعابير النّظامية في مرونتها لتوصيف هويّات التّطبيقات(application signatures) الّتي يتم البحث عن تطابق لها ضمن الرزم التي تعبر الشبكة إذْ إنَّ الهوية هي عبارة عن وجود مجموعة مركبة من العديد من السلاسل المحرفية المنفصلة عن بعضها وغير محددة المواقع والمنتشرة ضمن كامل الرزمة (أو الرزم) وإحدى الطرائق لنمذجة هذه التركيبة تتم باستخدام التعابير النظامية. حيث تستخدم في كثير من أنظمة التشغيل والبرمجيات على سبيل المثال أنظمة كشف الاختراقات الشبكية Snort و Bro وكذلك Cisco Adaptive Security وبرامج مكافحة الفيروسات ClamAV وأنظمة منع الرسائل غير المرغوب فيها SpamAssassin [2][3][4].

2 -3 تكييف دفق البيانات:

أسلوبٌ يستخدم للتحكّم في سير البيانات ضمن الشبكة بغية ضمان مستوى محدد من الأداء. يَنتج تكييف سيل البيانات عن تفعيل قواعد ضبط أرتال البيانات في المسيّرات يوضح الشكل (1) وصلة انترنت قبل تكييف الدفق للتطبيقات وبعده كما نلاحظ دون وجود آلية لتكييف الدفق تحاول التطبيقات الشرهة للسعات مثل برامج P2P الاستحواذ على الجزء الأكبر من سعة الوصلة ولكن مع وجود آليات تكييف الدفق يمكن ضبط السعات لكل تطبيق حسب الأولوية.



الشكل (1) تكييف - تشذيب - الدفق لوصلة إنترنت

مفاهيم تكييف الدفق وتتضمن مايأتي:

-تصنيف الرزم Classification: تُسنَدُ الرّزم إلى صفوف (classes) مختلفة حيث يجري البحث أو مطابقة الرزمة بقواعد ونماذج محددة من أجل إسنادها إلى أحد الصفوف.

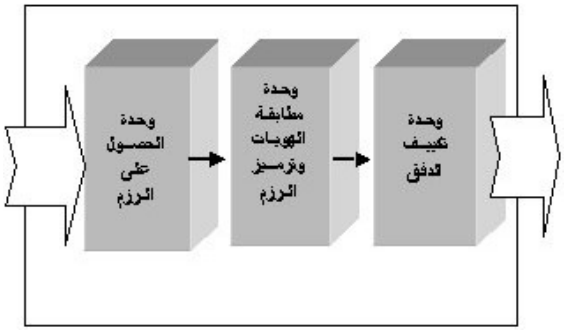
- الارتال queuing: يتألّف رتل الحزم من صوان (Buffer) يقوم بحفظ الرزم حتى يتجاوز حجم البيانات المستقبلة في المسيّر قدرته على الإرسال توضع الرزم في أرتال مختلفة بناء على الصفوف التي تنتمي إليها.

- الجدولة scheduling

وهي عمليّة اختيار الرزم الموجودة وإرسالها في الأرتال تبعاً لأولوية حزمة البيانات ووضع الرتل يوجد العديد من الخوارزميّات لتحقيق ذلك وتدعى آليات تنظيم الرزم في الأرتال ومنها HTB SFQ TBF RED [5]

3- التصميم الأولي للنظام وتحقيقه:

يوضح الشكل (2) المخطط العام للمنظومة المراد تصميمها.



الشكل (2) : المخطط العام لمكونات للنظام

نلاحظ أن هذه المنظومة تتألف من العناصر الأساسية الآتية:

- وحدة الحصول على الرزم.
- وحدة مطابقة الهويات وترميز الرزم.
- وحدة تكييف الدفق.

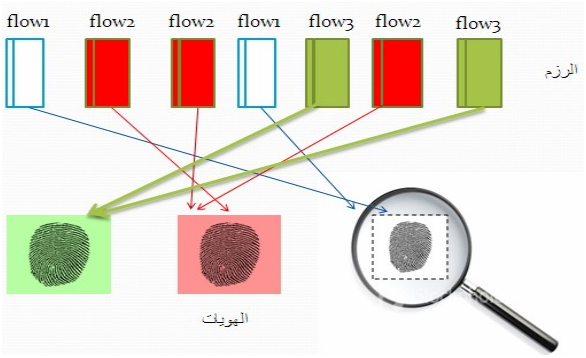
تُصنَّف لمنظومة على منصة حاسوبية تعمل كمسيّر للرزم من الشبكة وإليها بالاعتماد على نظام التشغيل Linux والوحدات البرمجية المفتوحة المصدر التي يوفرها وقد دُرِسَ العديد من تلك الوحدات التي تساعد في تحقيق المنظومة وآليات عملها وتكاملها مع بعضها الآخر واختيار المناسب منها مما يساعد في تحقيق منظومة البحث وتشكيلها.

3 -1 وحدة الحصول على الرزم:

في هذه الوحدة يتم تفعيل التسيير على المنصة الحاسوبية ضمن نواة نظام التشغيل وتوجيه الرزم المارة إلى رتل انتظار وذلك من خلال الاعتماد على البيئة Netfilter [6] كونها ذات بنية مستقرة وناضجة تقنياً التي توفر الأداة IPTABLES ومن ثمَّ حُوِّلَ مسار عبور الرزم ضمن نظام التشغيل إلى رتل الاصطفاف للرزم.

3 -2 وحدة مطابقة الهويات مع الرزم وترميز الرزم:

هذا الجزء هو برنامج في فضاء المستخدم (user space) يقوم باستقبال الرزم من رتل الاصطفاف للرزم في الوحدة الأولى وذلك باستخدام بعض الواجهات البرمجية المقدمة من البيئة netfilter بعد ذلك تقوم هذه الوحدة البرمجية بالبحث عن هويّة التّطبيق ضمن أعماق الرزم وفي حال حدوث تطابق للهوية نقوم بتغيير بعض القيم في ترويسة الرزم للدلالة على تصنيفها لتتابع بعدها المسير.



الشكل (3): البحث عن الهوية ضمن الرزم

المصنف L7-filter:

وهو المصنف الذي اعتُمِدَ ضمن هذه الوحدة من المنظومة ويعدُّ من أهم التّطبيقات المفتوحة المصدر في نظام التشغيل Linux التي تقوم بتصنيف للرزم وفقا للتطبيقات [7] تناول كثير من الباحثين هذا البرنامج من أجل دراسة موضعات البحث العميق داخل الرزم والبحث عن الهويات والتعابير النظامية.

مفهوم التدفق flow : هو الاتصال الشبكي بين طرفين ويتحدد بالخماسية: البرتوكول وعنوان المرسل وعنوان المستقبل والمنفذ المصدر والمنفذ الوجهة.

إنَّ لكل تطبيق تعبيراً نظامياً لمطابقته سوف يتم البحث عن هذا النموذج (pattern) في حِمْلِ البيانات (payload وهي المعطيات التابعة للتطبيقات و الموجودة في الرزم) الخاص بطبقة التطبيق وعملية البحث والتطابق هذه تبدأ من الرزم الأولى للتدفق ولعدد محّددٍ منها فليس من المُستحسن البحث أو مطابقة التعبير النظامي لكلِّ رزمة من الرزم التي تمرّ وإنّما يُكتفى بالرزم المتبادلة في بداية التدفق (لكلّ اتصال مختلف) إذْ إنَّ هذا كافياً لكشف كثير من التطبيقات (كون الهوية توجد عادة خلال المراحل الأولى في بداية الاتصال حيث تكون معلومات التحكم الخاصة بالتطبيق) وهذا ما نطلق عليه الحد الأعظم للرزم التي سوف تعالج ضمن كل اتصال لذلك يقوم المصنف بحفظ بيانات الحمل للرزم لكل اتصال ما دام لم يتجاوز عددها الحد الأعظم ومادام لم يحدث تصنيف للاتصال بعد لكي يتمكن من عمليات المطابقة على تلك البيانات في حال انتشار الهوية على عدة رزم.

تتم مطابقة التعابير النظامية بشكل افتراضي مع أو 2048 Byte من الرزم التابعة لتدفق ما (أو أول عشر رزم لأنَ الهوية عادة تكون مؤلفة من مجموعة من معلومات التحكم المتبادلة في بداية الاتصال) ومن الممكن التحكم بهذه القيم وتغييرها.

تُعالَجُ حالة الاتصال ضمن البرنامج l7-filter حيث يتم استدعاء المكتبات الخاصة الّتي تقدمها netfilter لتحقيق ذلك وهي libnetfilter_conntrack

وتُمَيَّزُ ثلاث حالات للاتصال الذي تتبع له الرزمة عند معالجتها ضمن البرنامج:

1 -الاتصال صُنِّفَ مسبقاً (التصنيف تم بشكلٍ صحيح أو تم تجاوز الحد الأعظم للرزم للاتصال نفسه Classified).

2 -الاتصال لم يُصَنَّفْ سابقاً وهو قيد التصنيف ولم يتجاوز عدد الرزم التابعة للاتصال الحد الأعظم بعد (وهي الحالة الّتي يتم فيها البحث عن التعبير النّظامي ضمن معطيات الاتصال No-Match-Yet).

3 -الاتصال لم يُصَنَّفُ مسبقاً ولكن مع الرزمة الحالية فسوف يتجاوز عدد الرزم التابعة لهذا الاتصال الحد الأعظم لذلك سوف تتوقف محاولة تصنيف كلّ الرزم التابعة لهذا الاتصال مستقبلاً (ويتم تصنيفه بقيمة افتراضية No-Match) .

ولكن لكي يتوقف البحث في الرزم التابعة لاتصال تم تطابقه مع نموذج (patterns) علينا تمييز الاتصالات وحالتها **وحفظ** تلك الاتصالات وحالتها في الذاكرة الحاسوبية للمنظومة وهذا ما يسمى نظام تتبع حالة الاتصال (connection tracking system).

وهو الإبقاء على حالة ومعلومات الاتصالات ضمن جداول خاصة في الذاكرة مثل العنوان الرقمي للمرسل والمستقبل وكذلك المنفذ المصدر والوجهة والبروتوكول وحالة الاتصال والقيم الزمنية للاتصال (timeout) وقيمة المميز الخاص بالاتصال في حالة تصنيفه.

عند حدوث تطابق للتعبير النظامي المقابل لهوية التطبيق مع بيانات الرزم لاتصال ما يُرَمَّزُ الاتصال بقيمة معينة أي تعليمه بتلك القيمة دلالة على تصنيفه يتم ذلك من خلال حفظ حالة الاتصال وقيمة ترميزه في الذاكرة وعند معالجة كل رزمة تتبع لذلك الاتصال تُوضَعُ قيمة المميز تلك ضمن الحقل nfmark الموجود ضمن بنية الرزمة(بنية الرزمة ضمن نظام التشغيل) وتتابع الرزمة مسيرها ضمن المكدس الشبكي لنظام التشغيل.

3 -3 وحدة تكييف التدفق:

بعد أن يتم تعليم الرزم بالمميز الخاص بالتطبيق في حال حدوث تطابق لا بدَّ من الاستفادة من ذلك يتم ذلك من خلال تطبيق أنظمة جودة الخدمة QoS إذْ إنّه بناءً على قيمة المميز الموجودة ضمن الرزمة (قيمة الحقل nfmark التي يتم إسناد قيمتها من قبل المصنف في الوحدة الثانية) يتم إدراج الرزم ضمن أحد صفوف كفاءة الخدمة المختلفة التي يُعَرِّفُها نظام جودة الخدمة.

في هذا المكون نستخدم الأداة (Tc) traffic control [8] لإعدادات التحكم بالدفق المار في نواة نظام التشغيل Linux ويتم التحكم بذلك الدفق من خلال الأغراض الآتية:

-QDISCS: وهي آليات تنظيم الرزم في الأرتال queuing discipline

-CLASSES: وهي صفوف الخدمة وقد تكون هرمية وفيها تعرف واصفات الصف مثل سعة الصف وغيرها.

-FILTERS: وهي المرشحات التي تستخدم لتحديد ما الصف الذي سوف تُسْنَدُ الرزمة إليه بناءً على قيمة المميز الموجود ضمن الرزمة.

4 - التجارب العلمية والنتائج:

في هذا الجزء قمنا بكتابة برمجيات اختبارية اعتماداً على بيئة netfilter لقياس أثر البحث عن الهوية ضمن أعماق الرزم في الأداء بأشكالها المختلفة أي إنَّ الهوية هي عبارة عن سلسلة محرفية ثابتة أو تعبير نظامي أو تمت نمذجتها باستخدام نموذج تطابق البايتات Byte Pattern تقوم تلك البرمجيات بمطابقة الهويات ضمن أعماق الرزم المارة عبر المسيّر وقياس المردود (السعة الانتاجية) ومن ثم زيادة عدد الهويات مرة أخرى وإعادة قياس النتائج لذلك قمنا بإعداد بيئة تجريبية مؤلفة من ثلاث منصّات حاسوبية المنصّة الأولى حاسوب عليه نظام التّشغيل Linux يعمل كمخدم ويب web server وتم تخزين ملفات كبيرة الحجم فيه من مراتب الغيغا بايت أمَّا المنصّة الثانية فهي حاسوب يعمل كمسيّر وتم إعداد البرنامج الاختباري عليه من أجل إجراء الاختبارات أمَّا المنصّة الثالثة فهي حاسوب عليه نظام التّشغيل Linux أيضاً يقوم بالاتصال بمخدم الوب (عن طريق المنصة الثانية) وتحميل أحد الملفات كبيرة الحجم (مع ملاحظة عدم وجود نظام تتبع حالة الاتصال لكي لا تتوقف المطابقة بعد عدد قليل من الرزم وبذلك نضمن أن

البرنامج الاختباري يقوم بعمليات المطابقة على كل رزمة تعبر المسيّر).

نلخص خوارزمية عمل البرمجيات الاختبارية كما يأتي:

1 - تمر كل الرزم من منصة الزبون - المخدم وإليه عبر المسيَر(المنصة الثانية).

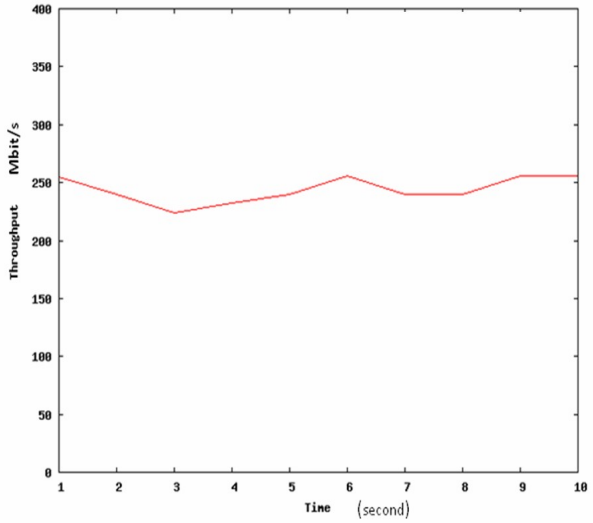
2 - في المنصة الثانية (المسيَر) نقوم بتحويل الطريق الاعتيادي للرزم التي تمر عبر المكدس الشبكي لنواة نظام التشغيل بحيث تُوَجَّهُ إلى رتل مؤقت.

3 - تقوم الوحدة البرمجية الاختبارية (اعتماداً على بيئة netfilter) بقراءة الرزم واحدة تلو الأخرى من الرتل بعد ذلك مطابقة حمل الرزمة مع الهوية بشكل تكراري إلى عدداً من المرات محاكاة لوجود عدد من الهويات المختلفة للتطبيقات.

4 - خلال ذلك نقوم بقياس المردود الحقيقي من المحطة الزبون إلى حين الانتهاء من نسخ الملف بعد ذلك نقوم بتكرار التجربة السابقة مع تغيير عدد الهويات التي تُطابَقُ مع حمل المعطيات لدراسة أثر عدد عمليات المطابقة في المردود.

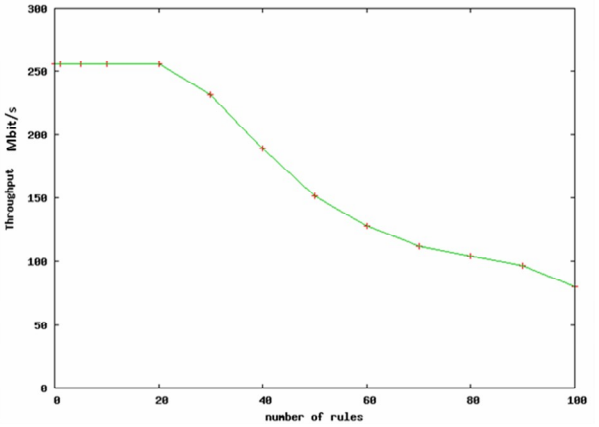
5 - نكرر الشروط السابقة ولكن مع تبديد أشكال الهويات المختلفة من البحث مع وجود هوية من نمط سلاسل محرفية ثابتة ومحددة والبحث مع وجود هوية عبارة عن تعابير نظامية والبحث مع وجود هوية من نمط مطابقات البايتات وذلك لدراسة أثر اختلاف أنواع تمثيل الهوية في المردود.

في الحالة العادية الّتي يقوم فيها المسيّر فقط بعمليّة تسيير الرزم دون إجراء أي عمليّات بحث ومطابقة حصلنا على النتيجة الّتي يوضحها الشكل (4) والّتي نلاحظ فيها استقرار المردود Throughput خلال عمليّة النقل مع مرور الزمن (تم الحصول على throughput من المحطة الزبون الّتي تقوم بتحميل الملف والّتي تظهرها أداة التحميل wget وهي تطبيق برمجي في نظام التشغيل للحصول على الملفات من مخدم الوب).



الشكل (4): معدل المردود دون عمليات بحث عن سلاسل محرفية أو مطابقة للتعابير النظامية

ولمعرفة تأثير عمليّات البحث ضمن الرزم كُرِّرَتِ التجربة السابقة ضمن الشروط نفسها لكن مع إضافة عدد من القواعد الّتي تبحث عن سلسلة محرفيه محددة وثابتة (ليست تعبيراً نظامياً) بواسطة خوارزمية Boyer-Moore (وهي إحدى الخوازميات الشهيرة للبحث عن السلاسل المحرفية)[9] وقياس المردود الموافق لذلك وتكرار التجربة مع زيادة عدد القواعد في كلّ مرة وحصلنا على النتائج الموضحة في الشكل (5) التي تشير الى الحفاظ على المردود دون تأثر إلى حين وصول عدد القواعد الى عشرين بعدها ينخفض المردود مع زيادة عدد القواعد.



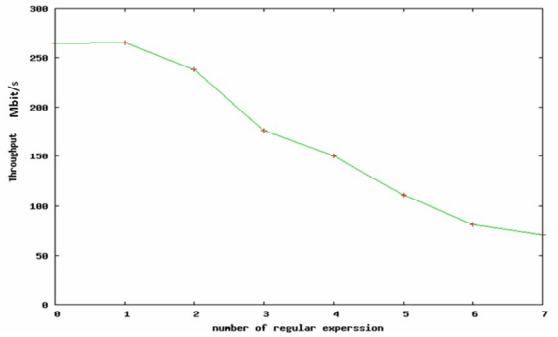
الشكل (5): معدل المردود مع عمليات بحث عن سلاسل محرفية ثابتة

وفي الدراسة التالية يتم البحث باستخدام التعابير النّظامية حيث تُوَجَّهُ الرزم باستخدام الأداة Iptables إلى رتل وكُتِبَ برنامج يقرأ الرزم من ذلك الرتل واستُدعي تابع يقوم بمطابقة تعبير نظامي ضمن الرزم وروقبَ الأداء وتكررت التجربة نفسها عدة مرات مع زيادة عدد الاستدعاءات وذلك لمحاكاة استدعاء أكثر من تطبيق إذْ إنَّ لكلّ تطبيق تعبيراً نظامياً -وسُجِّلَت النتائج المضمنة بالشكل(6) باستخدام التعبير النّظامي الّذي اختيرَ من المصنف l7-filter وهو من الشكل الآتي:

```
<stream:stream[\x09-\x0d][-~]*[\x09-
\x0d]xmlns=['"]jabber
```

وهو التعبير النظامي الذي يوصف البرتوكول jabber (ويعني هذا التطابق كل ورود لسلسلة المحارف stream:stream يليها وجود أحد المحارف الست عشرية بين المجال \x09\ و\x0d بعد ذلك أحد المحارف - ~ - صفر مرة أو أكثر - ومن ثم أحد المحارف الست عشرية في المجال \x09-\x0d يليها المقطع xmlns= بعد ذلك أحد المحرفين ' " ويليها المقطع jabber

نلاحظ من الشكل التالي (6) أنّه عندما كان عدد الاستدعاءات واحداً لم يتأثر الأداء تأثيراً كبيراً (نقصد بكل استدعاء مطابقة تعبير نظامي) ولكن بعد زيادة عدد الاستدعاءات إلى التعبير النّظامي نفسه بدأ الأداء يتأثر تأثيراً كبيراً وعند الوصول إلى 7 استدعاءات (أي مطابقة 7 تعابير نظامية) وصلنا إلى القيمة 70 ميغا بت في الثانية وهذا يظهر بشكلٍ واضح أن عدداً بسيطاً من التعابير النّظامية قد سبب انخفاضاً حاداً في الأداء من القيمة 264 ميغا بت في الثانية إلى 70 ميغا بت في الثانية.



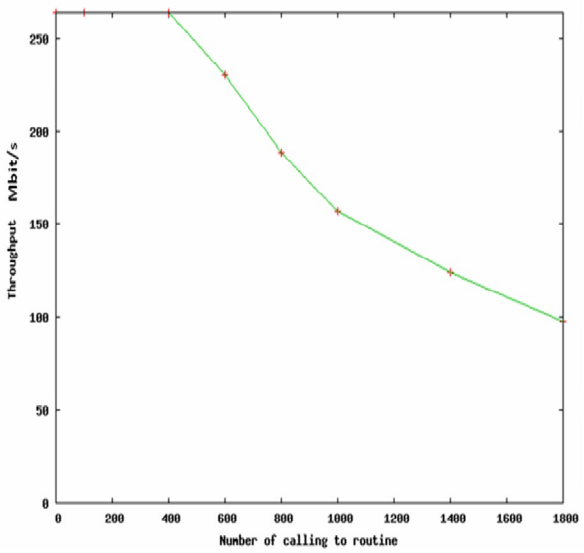
الشكل (6): معدل المردود مع عمليات مطابقة لتعبير نظامي لتوضيح أثر زيادة عدد عمليات المطابقة للتعابير النظامية إلى المردود

```
search_bittorrent (const unsigned char *payload, const u16 plen)
{
  if (plen > 20)
  {
    /* test for match 0x13+"BitTorrent protocol" */
    if (payload[0] == 0x13)
    {
        if (memcmp(payload+1, "BitTorrent protocol", 19) == 0) return (IPP2P_BIT * 100);
    }

    /* get tracker commandos, all starts with GET /
     * then it can follow: scrape| announce
     * and then ?hash_info=
     */
    if (memcmp(payload, "GET /", 5) == 0)
    {
        /* message scrape */
        if ( memcmp(payload+5, "scrape?info_hash=", 17)==0 ) return (IPP2P_BIT * 100+1);
        /* message announce */
        if ( memcmp(payload+5, "announce?info_hash=", 19)==0 ) return (IPP2P_BIT * 100+2);
    }
  }
  else
```

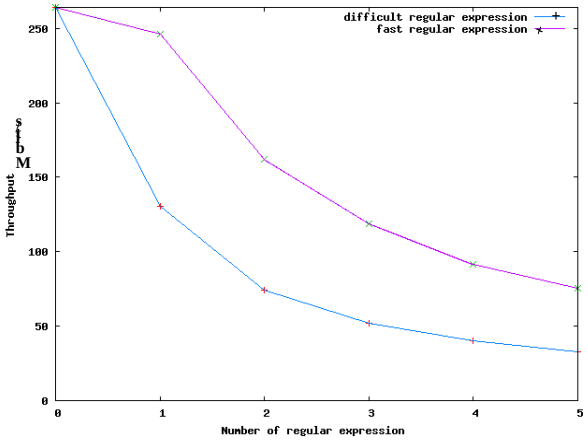
الشكل (7): مقطع من الرماز المستخدم في البرنامج ipp2p لتصنيف التطبيق Bittorrent

تَبيّن لنا في التجارب السابقة أثر عمليّات البحث والمطابقة في الأداء وفي تجربتنا التالية يتم اختبار الأداء على آلية التوصيف للهويّة الّتي يستخدمها البرنامج ipp2p [10] (أحد المصنفات الخاصة بتطبيقات الند للند) يستخدم البرنامج ipp2p طريقة أخرى للبحث عن هويّة التّطبيق ضمن الرزم الّتي تعتمد الوصول إلى أماكن محددة في الحمل ومقارنتها وكذلك بعض عمليّات المقارنة لطول الرزمة ومقارنتها بقيم محددة دون استخدام التعابير النّظامية مثل مايوضح الشكل السابق رقم(7) بعض الرماز المستخدم لتصنيف تطبيقات Bittorrent.



الشكل (8): معدل المردود باستخدام الرماز المستخدم في البرنامج ipp2p لتصنيف التطبيق bittorrent

كما يوضح الشكل (8) نلاحظ أنّه في أول 400 استدعاء لم يتأثر الأداء نهائياً (تقريباً) وحتى عند الوصول إلى 1000 استدعاء حافظنا على أداء بقيمة تتجاوز 100 ميغا بت بالثانية ويوجد فارق كبير جداً مقارنة بالنتائج التي حصلنا عليها باستخدام التعابير النظامية لتصنيف التطبيق نفسه، كما يوضح الشكل (9).



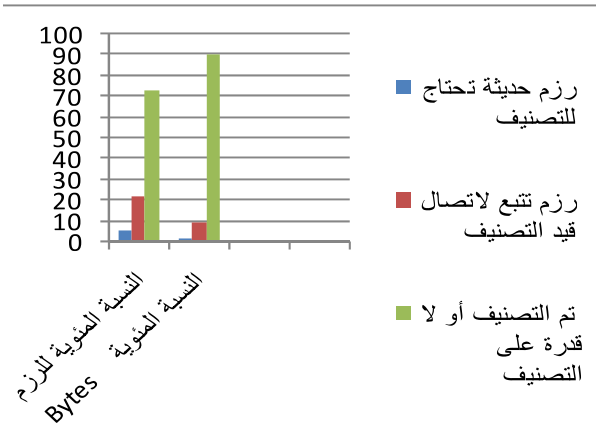
الشكل (9): معدل المردود باستخدام تعبيرين نظاميين مختلفين لتصنيف التطبيق نفسه bittorent لتوضيح أثر تعقيد كتابة التعابير النظامية في المردود

لدراسة أثر نظام تتبع حالة الاتصال الموجود ضمن المصنف كما فُصِلَ في الفقرة (3-2-) قمنا بتنفيذ الاختبار الآتي: ضمن شبكة مستخدمين تتجاوز 50 مستخدماً متصلة بشبكة الإنترنت عبر المسيّر الافتراضي للشبكة تم تنصيب محطّة حاسوبية قبل المسيّر الافتراضي لكي تمرَ عبرها الرزم جميعها من الشبكة الداخلية وإليها وباستخدام المصنف l7-filter على هذه المنصة من أجل تصنيف الرزم حُسِبَ عدد الرزم الّتي يعالجها البرنامج وحالاتها بلغ عدد الرزم التي عولجت ضمن الاختبار (21 010 533) رزمة ويتم تمييز ثلاث حالات عند معالجة الرزمة ضمن البرنامج:

1-الرزمة حديثة تحتاج إلى تصنيف.

2-الرزمة تتبع لاتصال قيد التصنيف (لم يحصل تطابق بعد ولم يتجاوز عدد الرزم المعالجة ضمن الاتصال قيد الحد الأعلى للرزم التي يتوقف عندها التصنيف لكل اتصال).

-الرزمة تتبع لاتصال صُنِّفَ سابقاً (مع ملاحظة أن الاتصال الذي أخفقَ تصنيفه بسبب تجاوز عدد الرزم قيد الحد الأعلى يعدُّ اتصالاً مصنفاً سلباً أي لا قدرة على تصنيفه) يلخص الشكل (10) النتائج



الشكل (10): نسبة الرزم وBytes باستخدام نظام تتبع حالة الاتصال

يظهر الشكل أهمية تتبع حالة الاتصال ضمن نظام المصنف حيث يتوقف التصنيف لكل اتصال بعد عدد محدد من الرزم وكما تظهر النتائج أن 73 بالمئة من الرزم ضمن بيئة الاختبار لن تُطابق مع التعابير النظامية لأنَّ اتصالاتها مصنفة مسبقاً (أو لا إمكانية لتصنيفها) مما يؤدي إلى تجنب العملية الأكثر كلفة واستهلاكاً لوحدة المعالجة في المنظومة وهي مطابقة التعابير النظامية ضمن الرزم العابرة.

5 - المناقشة:

عنق الزجاجة في منظومة البحث كما أثبتت التجارب السابقة هي عمليات مطابقة التعابير النظامية ضمن الرزم المارة [11] وهي عملية شرهة لوحدات المعالجة تفسر تلك النتائج بدراسة آلية عمل التعابير النظامية حيث تُعالجُ محارف الرزمة كلّها عند مطابقة التعبير النظامي معها محرفاً محرفاً ضمن آلة الأتومات المقابلة للتعبير النظامي إلى حين حدوث تطابق (إلا في بعض الحالات الخاصة التي يكون فيها التعبير النظامي محدداً بعملية تطابق مكاني في بداية السلسلة وإذا لم يحدث التطابق مع بداية السلسة تتوقف العملية)

لذلك وفق البحوث الحالية التالية تتم عمليات التحسين على العديد من المستويات لكلّ مكون من النظام وفق مايأتي:

5 -1 مناقشة التحسينات

أولاً: التحسين في التعابير النظامية

عملياً تُمَثَّلُ التعابير النظامية تُحَقَّقُ باستخدام نظريات الآلة: الأتومات الحتمي المنتهي DFA أو الأتومات اللاحتمي المنتهي NFA ولكلّ منهما تعقيده الزمني وحجم الذاكرة في مرحلتي البناء والمعالجة يمكن التحسين من خلال النقاط الآتية:

-إعادة كتابة التعبير النظامي بطريقة فعالة [12].

-حالياً ضمن الحلول البرمجية البحتة software-based تُستَخدَمُ الأتومات المحدد االلاحتمي NFA ولا يستخدم الاتومات المحدد الحتمي DFA لشراهته الكبيرة للذاكرة على الرغم من سرعة DFA الكبيرة مقارنة مع NFA لذلك تجري البحوث حالياً [13] على الاستفادة من السرعة الكبيرة التي يتمتع بها DFA مع محاولة الحد من مشكلة الذاكرة فيه لكي يُطَبَّقُ في هذه الأنظمة.

-أيضا تقوم البحوث الحالية بإيجاد حلول لمشكلة تمثيل التعابير النظامية ضمن نظريات الأتومات وذلك بتخفيف عدد الحالات والوصلات وكذلك إيجاد حلول هجينة تعتمد على DFA NFA [14]

ثانياً: التحسين في معالجة الرزم

-وضع قيد لعدد الرزم التابعة للاتصال نفسه التي سوف يتم البحث عن التطابق فيها.

-الاستغناء عن العمليات التي لا فائدة منها مثل الرزم المشفرة أو الاتصالات التي أخفقَ المصنّف بتصنيفها.

-الاتجاه السائد في التطبيقات الحالية من هذا النوع هو برمجتها بفضاء المستخدم user-space لذلك تنسخ الطرود من فضاء المستخدم-النواة وإليه وإذا استُخدِمَت القيود السابقة الذكر فلا داعي لنسخ الرزم التي لم تُصَنَّفْ أو تجاوز عدد الرزم لاتصالاتها القيمة العظمى ومن ثمَّ توفير كثيراً من وحدات المعالجة التي هي عنق الزجاجة في هذه الأنظمة مما يؤدي إلى التحسن الكلي في إنتاجية النظام .

ثالثاً: التحسين في البنية العتادية:

عنق الزجاجة في منظومة تصنيف الرزم هي توفر وحدات المعالجة التي يمكن المساعدة بحلها كما يأتي:

-استخدام بنى معالجات متعددة النواة تؤمن لنا التفرعيّة في إنجاز المهام فضلاً عن كتابة برمجيات بطريقة تستفيد من تلك التفرعية (باستخدام النياسب threads)[15].

-استخدام مسرّعات عتادية مثلا بطاقات معالجات برمجية خاصة تقوم بمطابقة التعابير النظامية وقد أصبحت موجودة فعلياً بشكل تجاري [16].

-الاعتماد على عتاد مادي بحت مثل دارت FPGA [17].

2-5 مناقشة الميزات والمحدوديات للأنظمة التي تعتمد على مطابقة الهويات:

-التعابير النظامية طريقة قابلة لتوصيف هويات التطبيقات ونمذجتها.

-الأنظمة التي تعتمد على التعابير النظامية تعطي نتائجَ أدق بكثير من التقنيات الأخرى.

-إيجاد التعابير النظامية عملية يدوية قد لا تكون واضحة ومباشرة تحتاج إلى التحديث والتعديل بشكل مستمر.

-الأنظمة التي تعتمد على التعابير النظامية لا تتعرف التطبيقات الجديدة بشكل آلي.

-التعابير النظامية وخصوصاً بشكلها البرمجي عملية شرهة لوحدات المعالجة والذاكرة.

-الخصوصية والسماحيات القانونية بالغوص إلى أعماق الرزم: إن التّطبيقات الّتي تعتمد البحث عن سلسلة محارف أو تعابير نظامية ضمن بيانات الرزم قد تشكل انتهاكاً للخصوصية وهناك بعض القوانين في العديد من البلدان تحدّ من ذلك لذلك يجب الأخذ بالحسبان عند محاولة اعتماد هذه التّقنيات لبناء تطبيقات أو منتجات شبيهة لمزودات خدمة الإنترنت أو حيث لا يمكن تطبيقها لاعتبارات الخصوصية أو الأمان للمعطيات وهذا ما يعطي أهمية للطرائق الأخرى الّتي لا تبحث عن الهويّة في أعماق الرزم. وإنما اعتماداً على الطرائق المَسلكية والإحصائية المُطبّقة على تدفقات البيانات العابرة.

-ضياع الرزم وتغير مسارها ضمن الشبكة: إن الرزم الّتي تنتقل في الشبكة قد تتعرض للضياع أو لا تحافظ على ترتيبها أو تسلك مسارات مختلفةٍ في الشبكة حسب التسيير المطبق في الشبكة وخصوصاً عند استخدام التسيير اللامتناظر (Asymmetric routing) وكما ذكر سابقاً فإنّ تصنيف الرزم يتم بناء على مطابقة بيانات الرزم مع هويّات التّطبيقات المختلفة سواءً كانت المطابقة تتم ضمن رزمة واحدة أو تمتد لعدة رزم وإذا ما حصل ضياع للرزم أو توجهت عبر عقدةٍ شبكية مختلفة عن العقدة الّتي تتم عندها عمليّة المطابقة فإنّ التّطابق لن يحصل ويخفقُ المصنف بعملية تصنيف للاتصال الذي بدوره يجعل عملية تشذيب (تكييف) الدفق بشكل صحيح غير ممكنة لهذا الاتصال.

-الأنظمة التي تعتمد البحث عن التعابير النظامية ضمن أعماق الرزم تصبح غير فعالة نهائياً في حالة استخدام التشفير والتغليف للرزم المارة أو تقنيات التشويش والجيل الجديد من برامج النّد للنّد يعتمد على ذلك ليتجنب أنظمة التصنيف ومن البرامج الشهيرة الّتي تستخدم ذلك أيضاً البرنامج الشهير skype ومن ثمَّ يجب البحث عن طرائق بديلة عن كشف الهويّة الّتي ركّز عليها البحث.

-تصنيف الاتصالات Related Session:

ضمن أنظمة التصنيف للرزم نواجه مشكلة في الاتصالات التي تسمى Related Session وهي الاتصالات التي يتم إنشاؤها من اتصال آخر عادة يكون اتصال التحكم مثلاً في البرتوكول FTP يولّد اتصال التحكم اتصالاً ثانياً لنقل المعطيات مختلف عن الاتصال الأول وكذلك فإنه من خلال بروتوكول SIP تتم إنشاء اتصالات RTP منفصلة.

فإذا لم تكن هناك قدرة على اكتشاف هوية لرزم تلك الاتصالات وخصوصاً أن تلك الاتصالات تحمل عادة معطيات مجردة للتطبيقات في حين معلومات التحكم تنتقل عبر الاتصال الأساسي سوف يخفق المصنف في تصنيف هذه الاتصالات.

ومن التقنيات الأخرى للتصنيف التي لا تعتمد على حمل الرزم هي الاعتماد على الطرائق الإحصائية وخوارزميات التجميع clustering التي تساعد في حل مشكلة التشفير والكشف الآلي للتطبيقات الحديثة [18].

تقنيات أخرى للتصنيف: لا تعتمد على التعابير النظامية.

-آلية مطابقة البايتات byte patterns تعتمد على مقارنة قيم (سلاسل) محرفية محددة وثابتة في عدة أماكن محددة ضمن أعماق الرزم وهي عملية غير مكلفة مقارنة بالتعابير النظامية لكنها تحتاج إلى دراسة دقيقة لرسائل التحكم للتطبيق وتفترض أن الأماكن التي سوف تُقَارنُ لا تتغير ضمن الرزم وإلا لم تعد الطريقة فعالة بينما في التعابير النظامية تحل تلك المشكلة حيث يتم البحث ضمن كامل الرزمة دون تحديد مواقع لكن على حساب الأداء.

6 - آفاق التطوير المستقبلية:

- استخدام المُسرِّعات العتادية المُتنوِّعة لتحقيق مُعدّل أداء من مراتب الغيغا بت في الثانية [16] .
- مطابقة التعابير النّظامية باستخدام نتائج البحوث الحالية[13][14] الّتي تعمل على أَمْثَلة وتحسين الأتومات المنتهي الحتمي واللاحتمي المُسْتَخدَم في تنفيذ التعابير النّظامية .
- الاعتماد على حلول هجينة من البحث عن التعابير النظامية فضلاً عن البحث عن سلاسل محرفية ثابتة ومُطَابقة البايتات(byte patterns) والتحليل السلوكي والإحصائي.

المراجع

[1] Subhabrata Sen, Oliver Spatscheck, and Dongmei Wang, “Accurate, scalable in-network identification of p2p traffic using application signatures” In Proceedings of the 13th international Conference on World Wide Web, Pages:512 – 521, 2004.

[2] Cisco Systems. Cisco Adaptive Security Appliance. http://www.cisco.com..

[3] "Snort", the. http://www.snort.org/.

[4] "SpamAssassin:Open-Source Spam Filter ", http://spamassassin.apache.org/.

[5] Leonardo Balliache, Differentiated Service on Linux, http://www.opalsoft.net/qos/DS.htm, 2003.

[6] Netfilter, Iptables, packet filtering framework, http://www.netfilter.org/

[7] L7-filter, Application Layer Packet Classifier for Linux,
http://l7-filter.sourceforge.net/

[8] TC, Linux man page, traffic control Tc, http://linux.die.net/man/8/tc.

[9] R. Boyer and J. Moore, A fast string searching algorithm, Communications of the ACM,

Pages: 762 –772, 1977.

[10] Ipp2p project, http://www.ipp2p.org.

[11] Niccol Cascarano, Luigi Ciminiera, Fulvio Risso, Computer Networks Group (NetGroup), “Accelerating DPI Traffic Classifiers”, 2009.

[12] Fang Yu, Zhifeng Chen, Yanlei Diao, T. V. Lakshman, Randy H. Katz, “Fast and memory efficient regular expression matching for deep packet inspection”, ANCS '06: Proceedings of the 2006 ACM/IEEE symposium on Architecture for networking and communications systems, 2006.

[13] Fang Yu, Zhifeng Chen, Yanlei Diao, T. V. Lakshman, Randy H. Katz, "Fast and memory efficient regular expression matching for deep packet inspection", ANCS '06: Proceedings of the 2006 ACM/IEEE symposium on Architecture for networking and communications systems, 2006

[14] Sailesh Kumar, Sarang Dharmapurikar, Fang Yu, Patrick James Crowley, Jonathan Turner," Algorithms to accelerate multiple regular expressions matching for deep packet inspection", Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, Pages: 339 - 350 , 2006.

[15] Danhua Guo, Guangdeng Liao, Laxmi N. Bhuyan, Bin Liu, Jianxun Jason Ding, “A Scalable Multithreaded L7-filter Design for Multi-Core Servers”, Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, Pages: 60-68, 2008.

[16] LSI Tarari Content Processor, http://www.lsi.com/networking_home/networking_products/tarari_content_processors/index.html.

[17] Abhishek Mitra, Walid Najjar, Laxmi Bhuyan ,“Compiling PCRE to FPGA for Accelerating SNORT IDS”, Proceedings of the 3rd ACM/IEEE Symposium on Architecture for networking and communications systems, Pages: 127-136, 2007.

[18] Jeffrey Erman, Martin Arlitt, Anirban Mahanti, “Traffic Classification Using Clustering Algorithms”, Proceedings of the 2006 SIGCOMM workshop on Mining network data, Pages: 281-286 , 2006.

تاريخ ورود البحث إلى مجلة جامعة دمشق 2010/6/21