Pengambilan Keputusan Pada Trafik *Management* Dengan Menggunakan *Reinforcement Learning*

Decision Making on Traffic Management Using Reinforcement Learning

Leonita Angelina[1], Yudha Purwanto[2], Astri Novianty[3]

[1,2,3]Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Telkom University
Bandung, Indonesia

[1]leonitangelina@students.telkomuniversity.ac.id, [2]omyudha@telkomuniversity.ac.id , [3]astrinov@telkomuniversity.ac.id

Abstrak

Seiring dengan berkembangnya kecanggihan dunia komunikasi, lalu lintas jaringan komputer juga semakin padat . Tidak menutup kemungkinan akan adanya serangan – serangan yang dapat menggangu trafik jaringan.Serangan yang sering kali dialami adalah serangan DDoS (*Distributed Denial Of Service*). Menurut [10] serangan *Distributed Denial Of Service* (DdoS) adalah jenis serangan yang dapat merugikan trafik jaringan yang sedang digunakan, baik terhadap target serangan maupun seluruh pengguna.

Setelah sistem mendeteksi anomali yang terjadi dan mengenali setiap serangan , maka management yang dilakukan berikutnya adalah pemilihan keputusan yang tepat untuk mengatasi serangan tersebut. Dalam kasus ini Reinforcement Learning dapat menjadi salah satu metode yang digunakan untuk memilih keputusan dalam penanganan anomali.

Hasil dari penelitian ini, sistem yang dibangun dapat bekerja dengan baik dalam pemilihan keputusan terbaik untuk mengatasi anomali trafik yang terjadi. Serangan yang terjadi akan dianalisa terlebih dahulu mneggunakan Reinforcement Learning dengan pemrosesan yang dibantu dengan Markov Decision Process .

Kata kunci : DDOS, *non agent, traffic shaping, QoS, token bucket filter*

Abstract

Along with the development of communication world, computer traffic's density has also increase. There is a possibility of serial network attack that can cause trouble of the network traffic. One of the attacks that frequently happen is DDoS. According to (10) DDoS is to cripple an online service by sending extremely large volumes of packets to a victim machine running the service.

After the system detect an anomaly and identify every attack, the next thing to do is making a right decision to handle the attack. In this case reinforcement learning can be applied as a method used to overcome the anomaly.

This result of this research, the system built can work well to overcome the anomaly. The attack happened will be analyzed first using reinforcement learning process also helped with Markov decision process.

1. Pendahuluan

Kemudahan dalam pengaksesan *data*, merupakan fasilitas awam yang bisa dirasakan menggunakan *internet* pada masa sekarang ini. Banyaknya user *internet* dari berbagai kalangan usia dan pekerjaanpun meramaikan lalu lintas jaringan *internet*. Para peretas pun semakin marak melakukan tindakan yang merugikan para user awam *internet*, yakni serangan yang dapat merugikan trafik jaringan yang sedang digunakan, baik terhadap target serangan maupun seluruh pengguna.

Sistem deteksi anomali trafik mempunyai kemampuan untuk mempelajari anomali setiap serangan sehingga dapat ditentukan penanganan yang maksimal untuk mengembalikan keadaan anomali yang dideteksi ke keadaan normal. *Survey* [4] menjelaskan bahwa jika terdapat data yang berada diluar *cluster* yang sudah diparsial maka bisa diidentifikasikan terdapat anomali ataupun intrusi.

Pengembangan sistem deteksi anomali memiliki banyak cara untuk mengetahui pola trafik normal sebagai acuan deteksi anomali trafik. Penelitian sebelumnya yang pernah ada yaitu penanganan anomali dengan mengetahui dulu karakteristik dari masing masing jenis serangan sehingga diketahui cara penanggulanannya. Berangkat dari hal tersebut Tugas Akhir ini mengunakan *Reinforcement Learning* yang mampu melakukan pemilihan keputusan yang paling tepat untuk mengatasi trafik yang terjadi pada lalulintas komunikasi dengan melakukan proses pembelajaran pada sistem.

2. Dasar Teori

2.1 Deteksi Anomali Trafik

Anomali trafik adalah suatu keadaan yang terjadi pada sebuah lalu lintas jaringan yang menyebabkan kondisi menjadi tidak normal. Anomali yang terjadi bisa dilihat melalui kenaikan lonjakan pengguna *internet*, melalui serangan pada suatu trafik dan lonjakan yang tidak disengaja. Kenaikan lonjakan dapat dilihat pada saat adanya bencana yang terjadi di dunia, kompetisi atau pertandingan dan kejadian yang tidak biasa terjadi setiap hari. Secara tidak sadar, kondisi kenaikan lonjakan ini memberikan dampak negatif bagi beberapa pihak. Kenaikan lonjakan yang terjadi tersebut menimbulkan penurunan performansi dari suatu jaringan. Untuk itu, perlu dilakukan deteksi terhadap anomali yang terjadi.

2.2 Distribution Denial of Services

Salah satu yang menyebabkan sebuah anomali pada trafik jaringan adalah serangan *DDoS*. Serangan ini biasanya dibuat dengan tujuan untuk melakukan pembekuan atau peniadaan hak akses dari sebuah komputer kedalam jaringan *internet*, sehingga korban yang diserang tidak dapat melakukan aktivitas pengaksesan apapun dalam layanan *internet*. Menurut survei [4], pada tahun 2012, *protocol* yang sering diserang oleh *DDoS* adalah port *ICMP, TCP, UDP*. *Protocol* ini digunakan untuk aktivitas *online* dan penyerangan yang dilakukan lebih banyak terhadap aktivitas *online* daripada *offline*.

Untuk melancarkan serangan *DDoS* seorang penyerang menggunakan *botnet* dan sebuah serangan sistem tunggal. Menurut sumber [5], *botnet* adalah sebuah aplikasi perangkat lunak yang dapat menjalankan tugas secara otomatis dalam *internet* dan menampilkan tugas yang berulang-ulang seperti mesin pencarian. *Botnet* digunakan untuk menyerang suatu *server* secara terus-menerus sampai *server* mengalami *down*, sehingga *server* tidak dapat diakses oleh *user* pada kurun waktu tertentu. Pada penelitian tugas akhir ini menggunakan *dataset* KDDCUP 99 sebagai *dataset* yang memiliki label *DDoS* [6].

2.3 Markov Decision Process

Markov Decision Process menurut pembahasan [3] Markov Decision Processes (MDP), adalah sebuah *framework* secara matematika yang di kembangkan oleh Andrey Markov untuk memodelkan sistem pengambilan keputusan dimana hasil dari keputusan adalah sebagian *random* dan sebagian di tangan pembuat keputusan. *MDP* sangat berguna dalam mempelajari banyak persoalan optimasi yang berhubungan dengan dynamic programming dan *Reinforcement Learning*.

2.4 Reinforcement Learning

Menurut Ali Barakbah, *Reinforcement Learning* adalah salah satu paradigma baru di dalam *learning* theory. *Reinforcement Learning* dibangun dari proses *mapping* (pemetaan) dari situasi yang ada di *environment* (*states*) ke bentuk aksi (*behavior*) agar dapat memaksimalkan *reward*. *Agent* yang bertindak sebagai sang *learner* tidak perlu diberitahukan *behavior* apakah yang akan sepatutnya dilakukan, atau dengan kata lain, sang *learner* belajar sendiri dari pengalamannya. Ketika *agent* melakukan sesuatu yang benar berdasarkan *rule* yang kita tentukan, *agent* akan mendapatkan *reward*, dan begitu juga sebaliknya.

Reinforcement Learning memiliki kemampuan mengadopsi proses *exploitation* dan *exploration* yang biasanya dilakukan oleh manusia. Kemampuan ini memampukan *Reinforcement Learning* bekerja berdasarkan pada informasi yang diterima sebelumnya dari aksi-aksi yang dilakukan pada waktu sebelumnya. Proses menggali informasi-informasi tersebut merupakan proses *exploitation*. Proses *exploration* adalah proses dimana diambilnya keputusan atau aksi tidak berdasarkan infomasi yang pernah terjadi sebelumnya melainkan melakukan aksi yang baru pertama kali akan dieksekusi. Untuk mendapatkan *reward* yang besar, agen *Reinforcement Learning* harus memilih *action* yang telah dicoba sebelumnya dan telah terbukti efektif untuk menghasilkan *reward* yang besar. Untuk menemukan *actions* tersebut, agen harus mencoba *action* yang belum pernah dipilih sebelumnya. Agen harus melakukan

eksploitasi terhadap apa yang telah diketahui dalam mendapatkan *reward* tetapi agen juga harus melakukan eksplorasi untuk menghasilkan pilihan *action* yang lebih baik di masa yang akan datang..

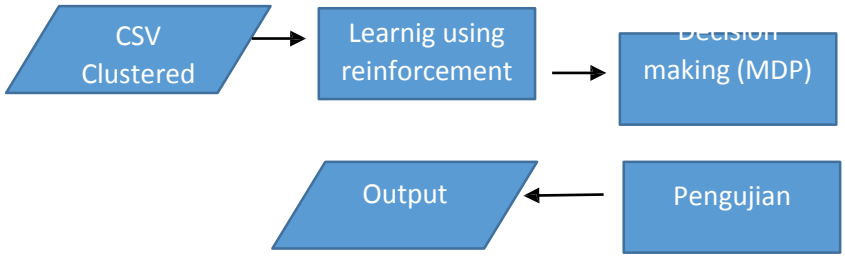
2.5 Kerangka Kerja Reinforcement Learning

"Permasalahan pada *Reinforcement Learning* sebenarnya merupakan permasalahan pada pembelajaran (*learning*) melalui interaksi untuk mencapai goal."[10]. Pada *Reinforcement Learning* yang disebut dengan agen adalah si pembelajar dan pembuat keputusan (*learner* and *decision maker*), sedangkan yang dimaksud dengan *environment* adalah segala sesuatu yang berinteraksi dengan agen dan berada di luar agen [7]. Proses interaksi berlangsung secara terus-menerus, dimana agen memilih *action* yang tersedia dan *environment* memberikan respon terhadap *action* tersebut serta memberikan situasi (*state*) baru kepada agen.

Environment juga memberikan *reward*, berupa data numerik, kepada agen ketika agen memilih sebuah *action*. Secara lebih spesifik, agen dan lingkungannya berinteraksi pada urutan waktu *(time steps)*, t = 0, 1, 2, 3, dst. Setiap *time step* t, agen menerima beberapa reperesentatif dari kondisi lingkungannya (*state*), s S t ∈ dimana S merupakan kumpulan *state* yang mungkin, dan berdasarkan *state* yang mungkin tersebut, agen memilih sebuah *action*, (t) a ∈ A s , dimana (t) A s merupakan kumpulan *action* yang tersedia pada *state* tersebut (t s). Sebagai akibat dari pemilihan *action* tersebut, agen menerima sebuah *reward*, r R t ∈ +1 , dan agen pindah ke *state* yang baru st+1.

3. Pembahasan

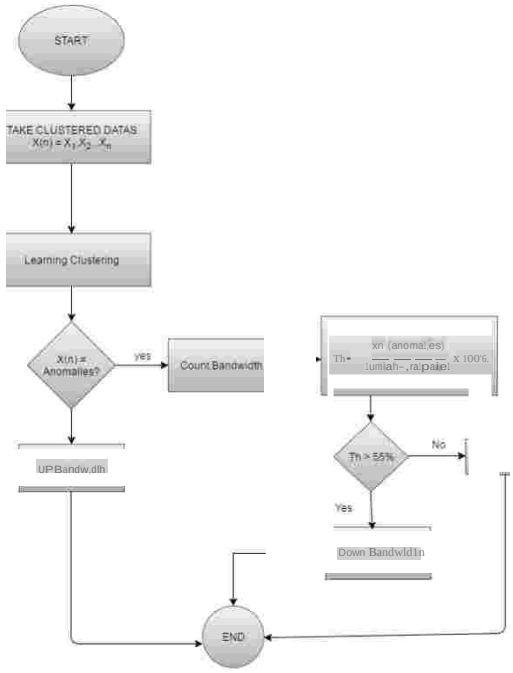
3.1 Deskripsi Sistem



Gambar 3.1 Blok Diagram Sistem

Berdasarkan gambar 3.1 dapat dijelaskan:

1. Alur sistem yang di representasikan pada gambar bagan pertama diawali dengan diterimanya data yang sudah di *cluster* terlebih dahulu sebelumnya menggunakan *DB-Scan* dan *Birch*.
2. Alur sistem dilanjutkan ke gambar bagan ke dua yakni proses pembelajaran dari sistem menggunakan *Reinforcement Learning* untuk pembelajaran penanganan jenis anomali yang terdeteksi.Sistem mulai bekerja apabila serangan sudah terdeteksi dengan status anomali sudah mencapai lebih dari sama dengan 55% dari total paket yang masuk.
3. Alur sistem bagan ketiga menjelaskan tentang pengambilan keputusan / langkah untuk mengembalikan anomali ke keadaan normal menggunakan *Markov Decision Process*.
4. Alur sistem bagan ke empat menjelaskan tentang apabila keadaan normal tidak tercapai maka sistem akan kembali lagi melakukan proses learning sampai keadaan normal tercapai.
5. Alur sistem bagan ke lima menjelaskan setelah keadaan normal tercapai maka proses sistem ini selesai.



3.2 Mengambil Data Clustering

Program dimulai dengan mengambil data hasil *clustering*, data hasil *clustering* ini berupa teks yang telah disusun menyerupai *dataset* KDDCup'99. Data ini nantinya dimasukkan kedalam *array*. *Array* tersebut nantinya diproses pada tahap selanjutnya.

3.3 Learning Hasil Clustering

Data yang telah diterima dari hasil clustering akan dipelajari oleh system, untuk mengetahui berapa persen dari jumlah *bandwidth* jaringan yang terserang anomaly dan berapa persen dari jaringan *bandwidth* yang masih normal. Hal ini dilakukan dengan melakukan perhitungan sebagai berikut :

$$P(\text{Anomali}) = \frac{\text{Jumlah Data Anomali}}{\text{Total Data}} \times 100\ \%$$

$$P(\text{Normal}) = \frac{\text{Jumlah Data Normal}}{\text{Total Data}} \times 100\ \%$$

Dari perhitungan tersebut, diketahui berapa persentase normal dan persentase anomali dalam total lebar *bandwidth* jaringan. Apabila terdapat serangan dengan status persenan serangan melebihi atau sama dengan 55 % maka proses normalisasi harus dilakukan.

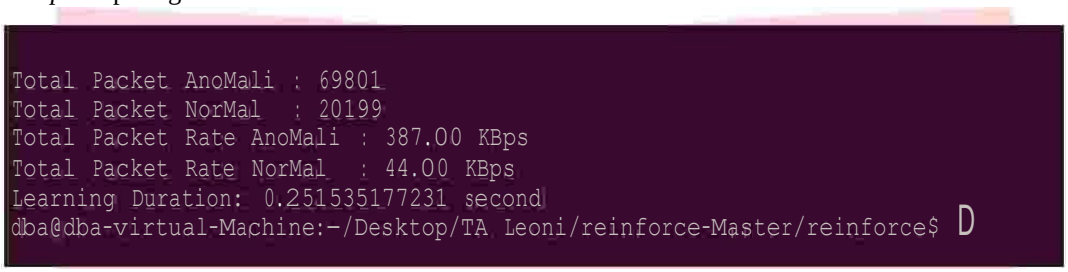
Untuk melakukan *down bandwidth* hanya dalam kondisi jumlah *bandwidth* jaringan sudah terisi lebih dari atau sama dengan 55%. Proses ini dilakukan dengan *learning* dalam beberapa kali iterasi atau *episode*. Hal sebaliknya dilakukan apabila bandwidth tidak mengalami serangan dengan intensitas melebihi 55%. Aksi yang akan dilakukan adalah *hold* ataupun *up bandwidth*.

.

4. Pengujian

4.1 Pengujian Pengambilan Keputusan

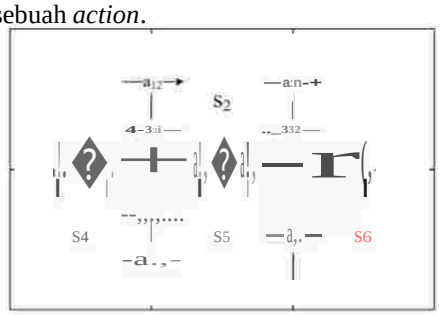
Data inputan dari hasil data yang sudah di *cluster* tersebut dilakukan proses learning sebanyak 200 kali iterasi dalam proses pembelajarannya dan dengan MDP dilakukan perhitungan probabilitas setiap perpindahan aksi nya sehingga diperoleh *output* seperti gambar :



Gambar 2.1 Output Decision

4.2 Pengujian Pengambilan Keputusan

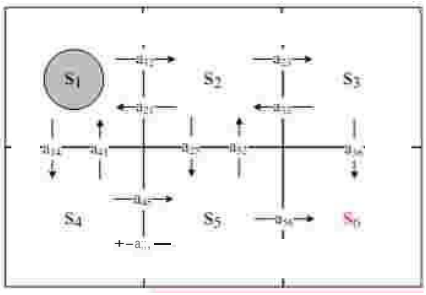
Pada kasus ini terdapat 6 buah state (*s*1, *s*2, *s*3, *s*4, *s*5, dan *s*6) yang direpresentasikan dalam bentuk *grid world* dan sebuah *action*.



Dari gambar di atas, *s*6 merupakan *goal state*, sehingga *reward* yang didapatkan oleh agen ketika mencapai *goal state* (*s*6) adalah 1, sedangkan yang menuju *state*
lainnya (*s*1, *s*2, *s*3, *s*4, dan *s*5) mendapatkan *reward* 0. Parameter γ yang digunakan adalah 0.5. Mula-mula *function table Q* diinisialisasikan dengan nilai 0 (agen belum memiliki pengetahuan). Selama proses *training*, agen akan meng-*update function table Q* untuk merekam pengalaman agen dalam setiap episode sehingga pada akhirnya agen dapat memperoleh optimal *policy*. Dalam setiap episode, agen akan memilih *state* awal secara acak (*random*) dan diperbolehkan untuk memilih action sampai mencapai *goal state*.

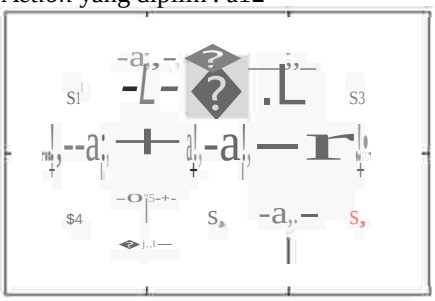
Sebelum proses *training* dijalankan, semua nilai *Q-value* diinisialisasi dengan nol. Seiring dengan berlangsungnya *training*, nilai-nilai ini akan terus di-*update* sehingga nantinya akan menghasilkan nilai *Q-value* yang konvergen.

-Episode 1 :



s1, a12	0
s1, a14	0
s2, a21	0
s2, a23	0
s2, a25	0
s3, a32	0
s3, a36	0
s4, a41	0
s4, a45	0
s5, a54	0
s5, a52	0
s5, a56	0

Posisi awal : *S*1
Pilihan action yang tersedia dari *S*1 : a12 dan a14
Action yang dipilih : a12



s1, a12	0
s1, a14	0
s2, a21	0
s2, a23	0
s2, a25	0
s3, a32	0
s3, a36	0
s4, a41	0
s4, a45	0
s5, a54	0
s5, a52	0
s5, a56	0

Posisi sekarang : S_2

Pilihan action yang tersedia dari S_2 : a_{21}, a_{25}, dan a_{23}

Update $Q(s_1, a_{12})$:

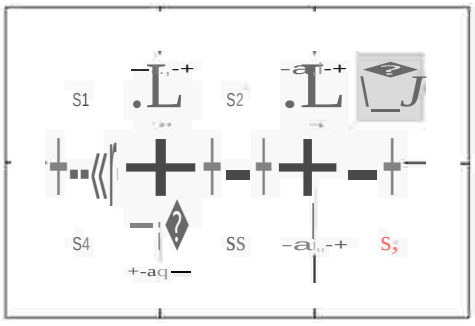
$Q^{\wedge}(s_1, a_{12}) = r + \gamma \max [Q(s_2, a_{21}), Q(s_2, a_{25}), Q(s_2, a_{23})]$

$Q(s_1, a_{12}) = 0 + 0.5*0$

$\wedge$

$Q(s_1, a_{12}) = 0$

Action yang dipilih : a_{23}



s1, a12	0
s1, a14	0
s2, a21	0
s2, a23	0
s2, a25	0
s3, a32	0
s3, a36	0
s4, a41	0
s4, a45	0
s5, a54	0
s5, a52	0
s5, a56	0

Posisi sekarang : s_3

Pilihan action yang tersedia : a_{31}, dan a_{36}

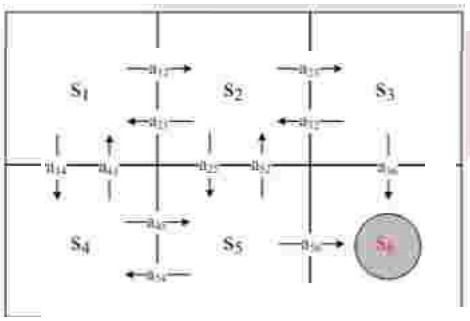
Update $Q(s_2, a_{23})$:

$$Q(s2, a23) = r + \gamma \max [Q(s3, a32), Q(s3, a36)]$$

$$Q(s_2, a_{23}) = 0 + 0.5*0$$

$$Q(s_2, a_{23}) = 0$$

Action yang dipilih : a_{36}



Posisi sekarang : s_6

Update $Q(s_3, a_{36})$:

s1, a12	0
s1, a14	0
s2, a21	0
s2, a23	0
s2, a25	0
s3, a32	0
s3, a36	1
s4, a41	0
s4, a45	0
s5, a54	0
s5, a52	0
s5, a56	0

Setelah sekian banyaknya iterasi, maka *table Q* yang terbentuk adalah:

s1, a12	0.25
s1, a14	0.25
s2, a21	0.125
s2, a23	0.5
s2, a25	0.25
s3, a32	0.25
s3, a36	1
s4, a41	0.125
s4, a45	0.5
s5, a54	0.25
s5, a52	0.25
s5, a56	1

Dari studi kasus di atas terlihat bahwa system / agent mampu belajar dan mengoptimalkan policy dari reinforcement learning. Hal ini bisa di lihat dari apabila state yang di ambil semakin menjauhi target (S 6) maka value yang akan di dapat akan semakin jauh dari 1.

5.Kesimpulan dan Saran

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian Tugas Akhir ini :

1. Penanganan serangan *DOS* dapat dilakukan dengan menggunakan *Reinforcement Learning*.

2. *Reinforcement Learning* mampu mengeksplorasi dan mengevaluasi dalam proses pembelajarannya untuk memperoleh keputusan terbaik dalam manajemen trafik.
3. Semakin banyak iterasi yang dilakukan maka nilai *value tree* akan semakin mendekati nilai optimum.
4. Implementasi pembelajaran *Reinforcement* mampu menentukan lajur yang efisien untuk mendapatkan goal *state* pada manajemen trafik yakni menurunkan *bandwidth* sebesar 387 KB dengan jeadaan jumlah paket anomaly yang terdeteksi adalah 69801 paket dari 90000 total paket.

Saran untuk penelitian selanjutnya adalah :

1. Program ini dirancang dengan menggunakan Bahasa *Python* dan menggunakan NS2 sebagai simulasinya, maka trafik yang digunakan untuk menjalankan program ini belum *streaming trafik*, hanya menyerupai saja. Diharapkan pada penelitian berikutnya bisa menggunakan simulator yang memang bias melakukan *streaming traffic*.
2. Pada penelitian ini, hanya menggunakan dua aksi yaitu *Up* dan *Down* saja. Diharapkan dalam penelitian berikutnya terdapat jumlah aksi yang lebih banyak untuk kedinamisan pemberian *reward* dan *punishment* dalam *learning process*.

DAFTAR PUSTAKA

[1] Y. Purwanto, Kuspriyanto, Hendrawan and B. Rahardjo, *Survey* : Metode dan Kemampuan Sistem Deteksi Anomali Trafik, Universitas Telkom 2015.

[2] K. P. Adiguna, Logika Fuzzy Untuk Menentukan Tindakan Penanggulangan Pada Jaringan, Bandung: Universitas Telkom, 2016.

[3] S. Jin, D. S. Yeung and X. Wang, "Network Intrusion Detection in Covariance Feature Space," *Pattern Recognition*, pp. 2185-2197, 2007.

[4] Akamai and PROLEXIC, "AKAMAI'S *STATE* OF THE *INTERNET*," 2013. [Online]. Available: www.prolexic.com. [Accessed 5 June 2015].

[5] E. Council, "Denial of Service," in *Ethical Hacking and Countermeasures v7.1*, 2011, pp. 433-440.

[6] A. T.U.K., "KDD CUP 1999 Data," 28 October 1999. [Online]. Available: http:/kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. [Accessed 21 April 2015].

[7] Ronald J.Williams, "*Reinforcement Learning* and *Markov Decision Process*es," CSG220,Spring 2007

[8] R.S Sutton,A. Barton, *Reinforcement Learning* : an Introduction , second printing,1999,The MIT Press.

[9] Tao Peng,Christoper Leckie, Kotagiri Ramamohanarao, "Proactively Detecting Distributed Denial of Service Attacks Using Source IP Addres Monitoring ," in Networking 2004,Springer Be*RL*in Heidelberg,2004

[10] Kleantis Malialis, "Distributed *Reinforcement Learning* fpr Network Intrusion Response,' University of York Computer Science , September 2014

[11] Ming Li , And,"Experimental Study Of DDoS attacking of flood type based on NS2, IJEC , December 2009