A Simulated Environment for Reinforcement Learning Based Intrusion Detection Using OMNET++

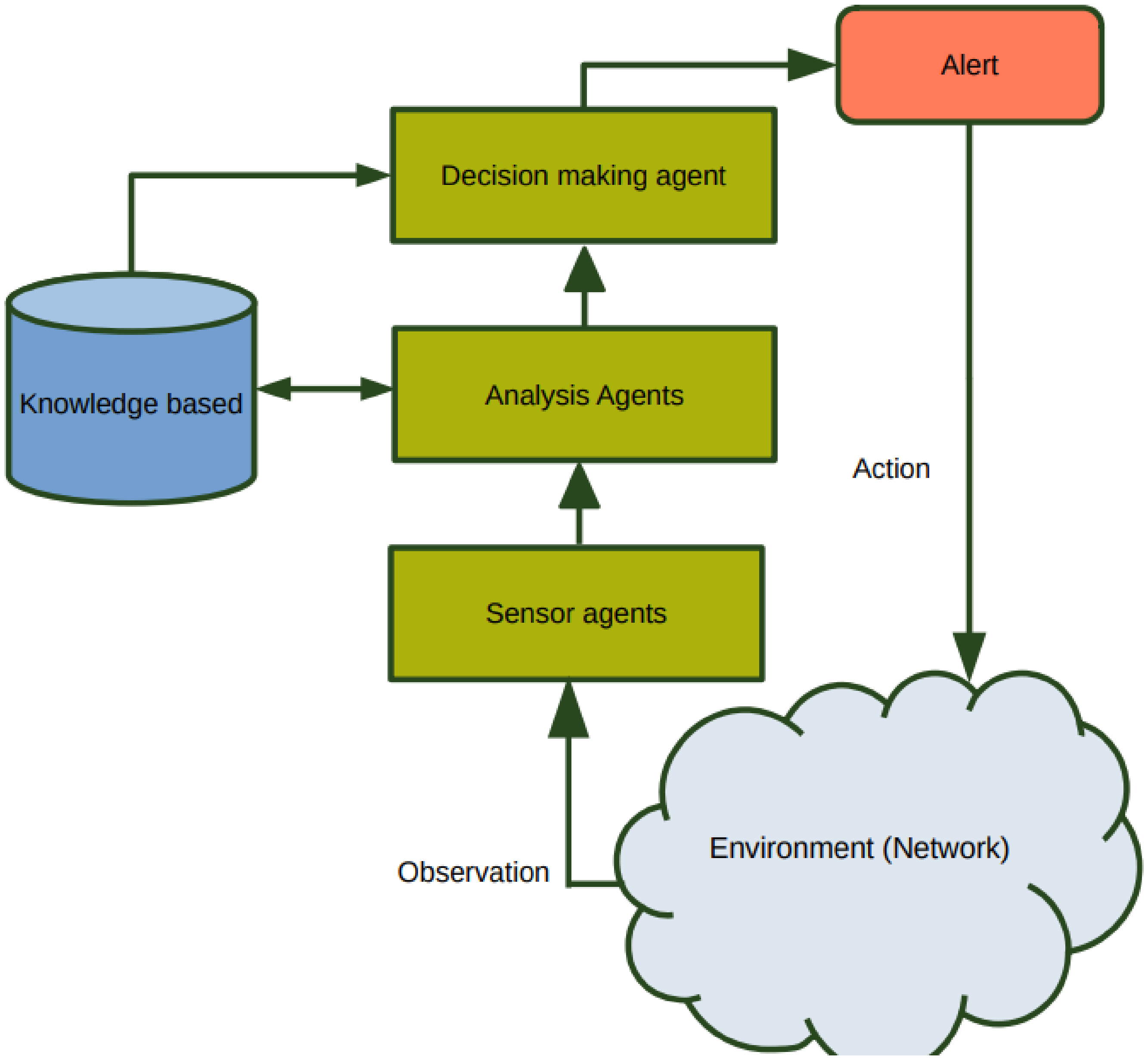
IMTITHAL ALNOUR SAEED

Agenda

- **Reinforcement environment.**
- **Proposed network environment.**
- **OSI Layers.**
- **Network data that sensed by the sniffer agents.**
- **Parameters For The HTTP Browsers and Servers**
- **DQN Agent (Detector).**
- **Network Traffic Before and After Attack using the DQN Agent.**
- **Performance of the DQN Algorithm (Q Value, Loss, and Epsilon)**
- **Detection Performance.**
- **Conclusion and Future Work.**

Reinforcement Learning Environment

- **Reinforcement Learning (RL) environment needs to be interactive.**
- **An gent learns by on-line interaction with its environment.**
  1. **Takes actions**
  2. **Receives feedbacks**
  3. **Evaluates the actions**
  4. **Updates their knowledge**
  5. **Repeats steps 1-4 until convergence**

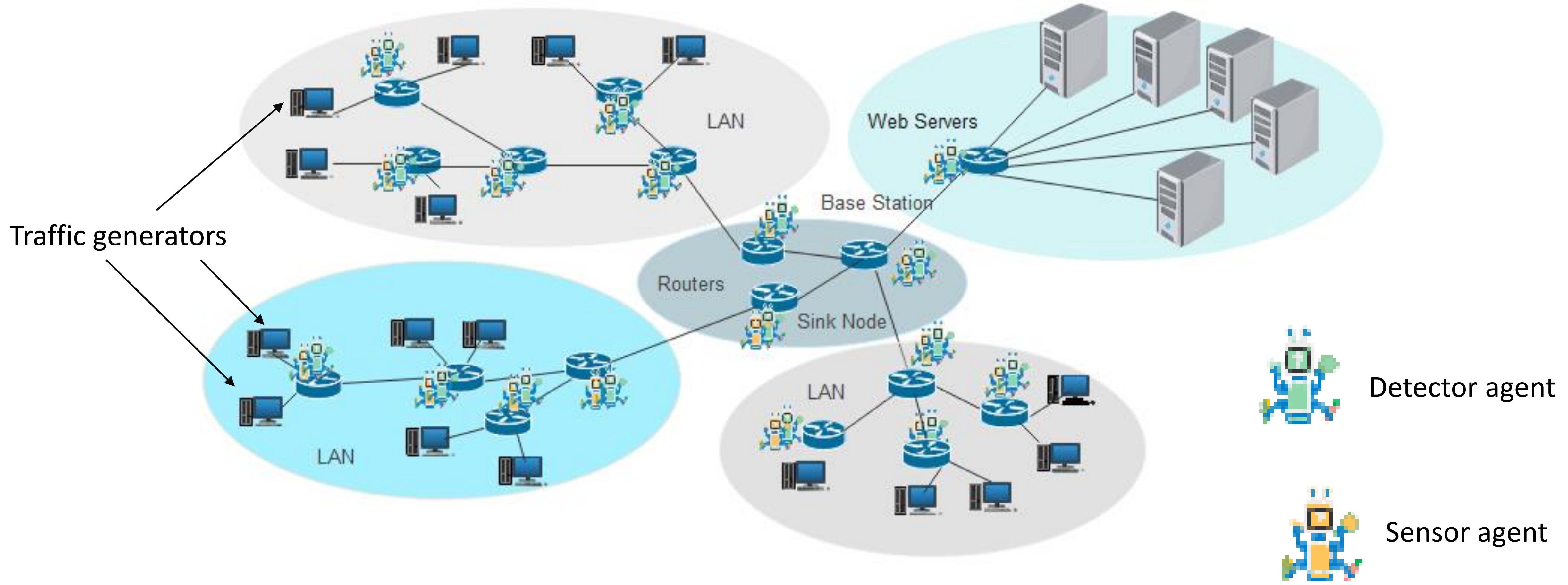


Reinforcement Learning Environment

- A sniffer agent function is embedded in the Data Link Layer.
- A sniffer agent continuously monitors the network state and feedback.
- An executer agent executes actions that change the network state .

Simulated Network

Proposed Network Environment



Proposed Network Environment

- HttpTools Kit of OMNET++ [1] has been utilized in developing the simulated network
- Standardhost components contain HTTP Browsers application for generating normal traffic.
- Servers are hosts but contain HTTP Servers applications to respond to requests sent by HTTP Browsers.
- Router are used for forwarding and dropping packets.
- Routers also host detector agents.
- HttpServerEvilA and HttpServerEvilB for generating malicious traffic.

Description of The Simulated Network Configuration

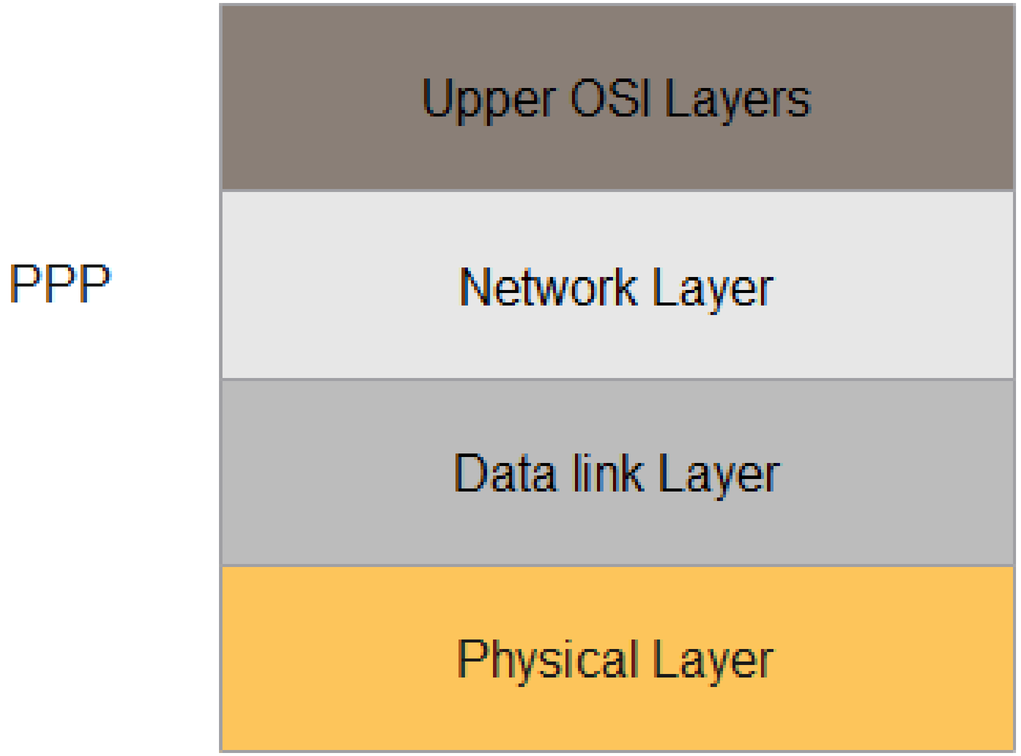
Simulated network configuration

General		
Service/protocol	Name	Description
Link	Ethernet links	Ethernet connection channels with capacity 100Mbps
Lookup service	HTTPControllar	Central service uses uniform popularity distribution to select servers
Web protocol	HTTP Tools kit	HTTP version 11
Devices specifics		
Device	Number	Description
Server	3	Uses HTTPServer
Host	11	Uses HTTPBrowser
Software specifics		
Name	Parameter	Description
HTTPServer	Start time=0s	Other parameters are specified by the HTTP Tool kit
HTTPBrowser	2 in each host	Parameters are specified by the HTTP Tool kit
HttpServerEvilA	number=1, start=0s	Parameters are specified by the HTTP Tool kit
HttpServerEvil	number=1, start=0s	Parameters are specified by the HTTP Tool kit

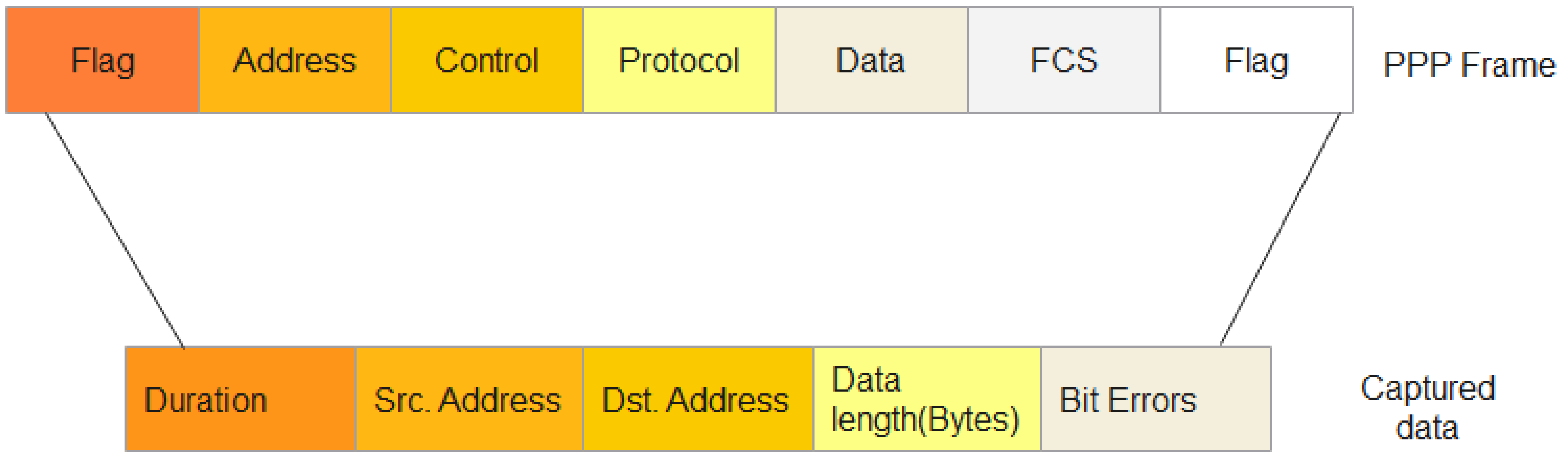
OSI Layers

OSI Layers

- A sniffer can see packets details after encapsulation.
- Captures relevant information
- Determines the network status based on a threshold.
- Communicates with the corresponding detector agent (on the same router)



Network State(Data) Sensed by Sniffer Agents



Network State(Data) Sensed by Sniffer Agents

Features Generated by the Sensor Agents

Features	Description
IP Addresses	Sources and destinations IP addresses
Data Transferred	Data transferred during a window of time equals 5sec
Number of requests	The number of requests during a window of time equals 5sec
Duration	The time between two recorded flows
Number of bit errors	The number of the bit error during a window of time equals 5sec

Parameters For The HTTP Browsers and Servers

Parameters For The HTTP Browser Application

HTTPBrowser Paramaters

Parameter	Description	Value
sessionInterval	The interval between activity periods	Normally distributed, μ = 3600, σ =1800s
requestInterval	The interval between requests within activity periods	Normally distributed, μ = 600, σ =60s
reqInSession	The number of requests per activity period	Normally distributed, μ = 10, σ =5s

Parameters For The HTTP Server Application

HTTPServer Paramaters

Parameter	Description	Value
pageSize	The number of referenced resources per HTML page	exponential, mean 10000 bytes, min=1000
numResources	The number of referenced resources per HTML page	uniform [0, 20]
textImageResourceRatio	The ratio of images to text resources on HTML pages	uniform [0.2, 0.8]
imageResourceSize	The size of image resources	exponential, mean 20000bytes, min=1000
textResourceSize	The size of text resources, e.g., CSS documents	exponential, mean 10000, bytes, min=1000

Parameters For The HTTP ServerEvilA and HTTP ServerEvilB Applications

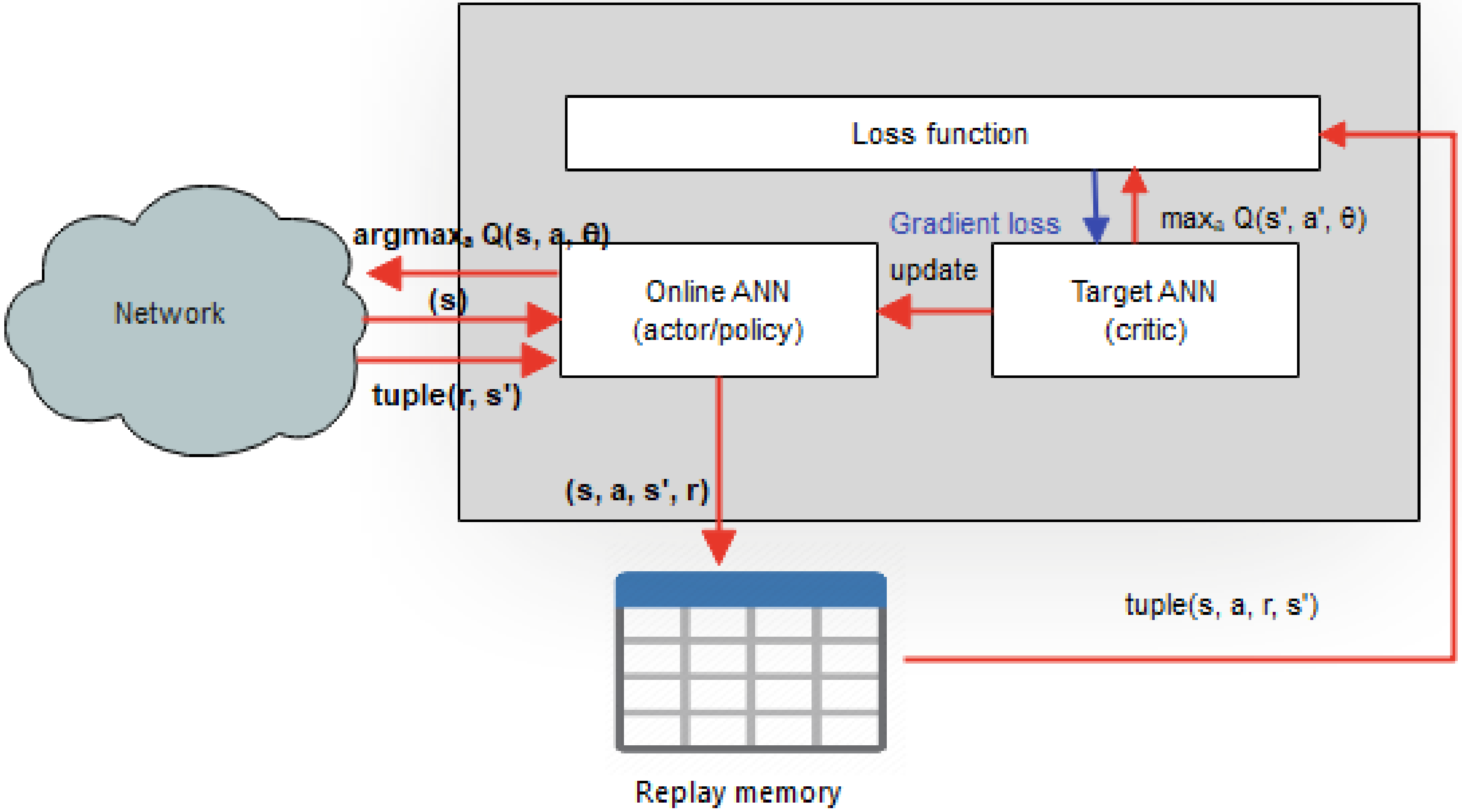
HttpServerEvilA and HttpServerEvilB Paramaters

Parameter	Description	HttpServerEvilA	HttpServerEvilB
minBadRequests	minimum requests number	3	3
maxBadRequests	maximum requests number	8	8

.

DQN Agent (Detector)

DQN Agent Architecture



DQN Agent Algorithm

Algorithm 1: DQN Algorithm for Single Agent Model

set replay memory m to size n
set random weights to $online$ network
set random weights to $target$ network
set episodes= e and interval=T
for $episodes = 1, 2, \ldots e$ **do**
  Set $s_1 = x, x = current\ \ network\ \ state \phi_1 = \phi(s_1)$
  for $t = 1, 2, \ldots T$ **do**
    Select action $a_t\ \ using\ \ \epsilon - greedy$
    Or select action with $\max_a Q^*(\phi(s_t), a, \theta)$
    Observe r, s_{t+1}
    Save $tuple(\phi_t, a_t, r_t, \phi_{t+1}) inm$
    Check if memory $m.size >= limit$ otherwise continue
    Select random sample from m #select a minibatch
    for $i = 1, 2, \ldots size(minibatch)$ **do**

$$y_j = \begin{cases} r_j & if \phi_{t+1} = terminal\ \ state \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a', \theta) & if \phi_{t+1} = non-terminal\ \ state \end{cases}$$

      Calculate $\nabla = (y_j - Q(\phi_j, a_j, \theta))^2$
      Update weights of both networks using ∇
      Accumulate $Q(\phi, a, \theta)$
    end for
  end for
end for
Use model for classification
for each tuple in dataset **do**
  Find an action with $\max_a Q^*(\phi(s_t), a, \theta)$
end for
Find true and false classifications
Evaluate model

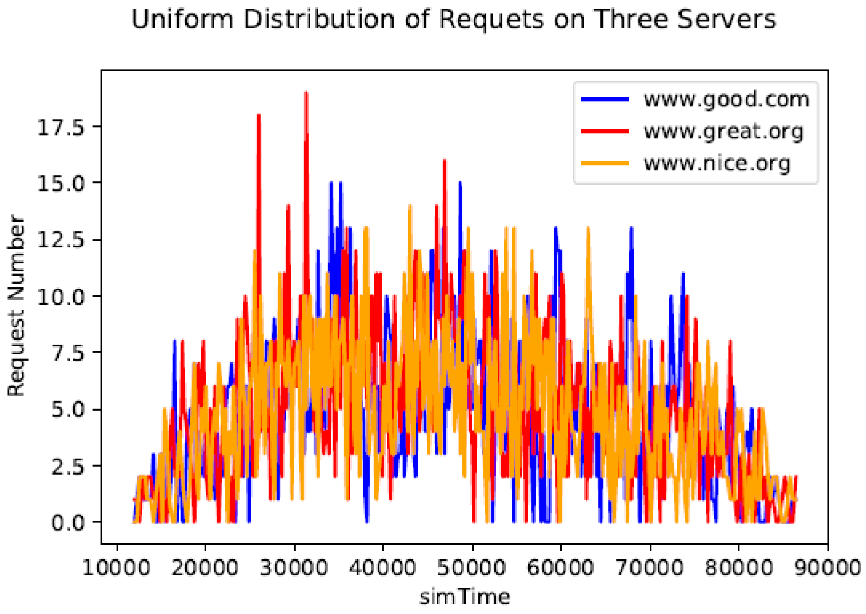
Parameters used with the DQN agent

Parameters used with the DQN agent

ANN Parameters		
Parameter	Value	Description
layers (deep)	4	The number of layers of the ANN
Inputs (s)	3	The number of inputs of the ANN
Outputs (a)	2	The number of actions of the ANN
eta (η)	0.01	Is used as a training rate for the ANN
alpha (α)	0.05	Is used as a training rate for the ANN
DQN Parameters		
Parameter	Value	Description
Input (s)	3	The number of inputs of the DQN
Actions (a)	2	The number of actions of the DQN
lr (α)	0.01	Learning rate of the DQN
gamma (γ)	0.05	Discount factor for the DQN
Minibatch size	20	The memory size the agent samples for training
epsilon (ϵ)	1	Probability used for balancing between exploration and exploitation
min-epsilon	0.90	minimum limit for epsilon
decay-epsilon	0.995	decay rate for epsilon

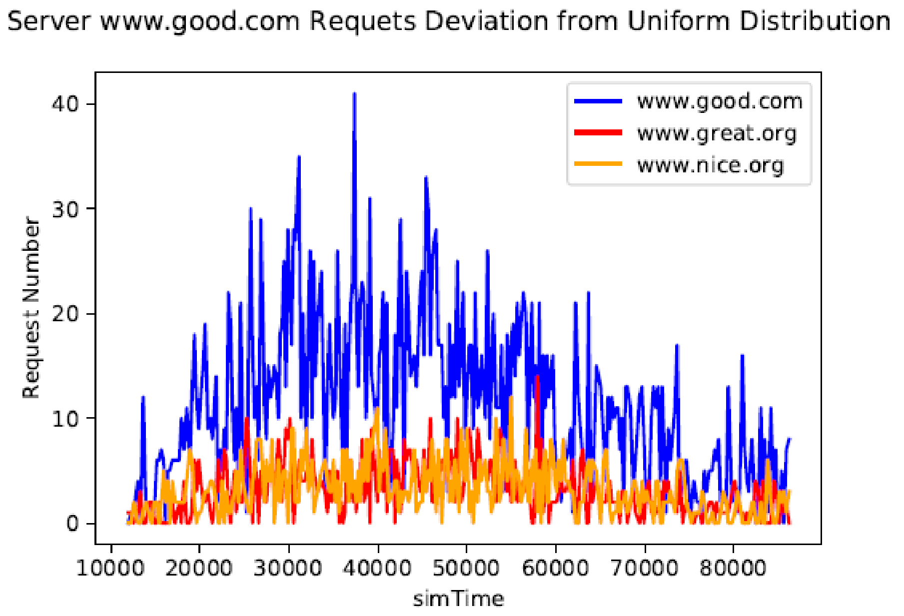
Network Traffic Before and After Attack using the DQN Agent

Three Servers in A Normal Situation



Uniformly distributed requests among servers in a normal situation

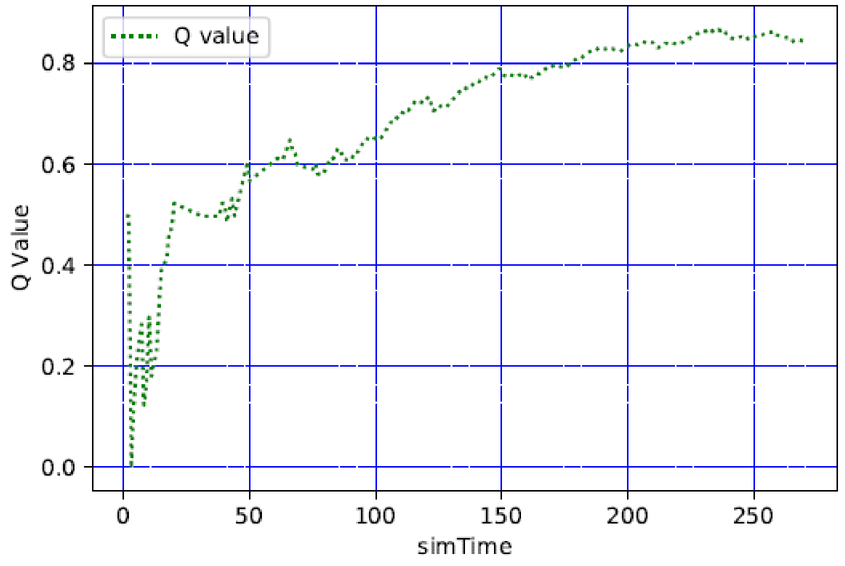
www.goog.com Server Under Attack



Comparing between servers’ traffic statuses under attack

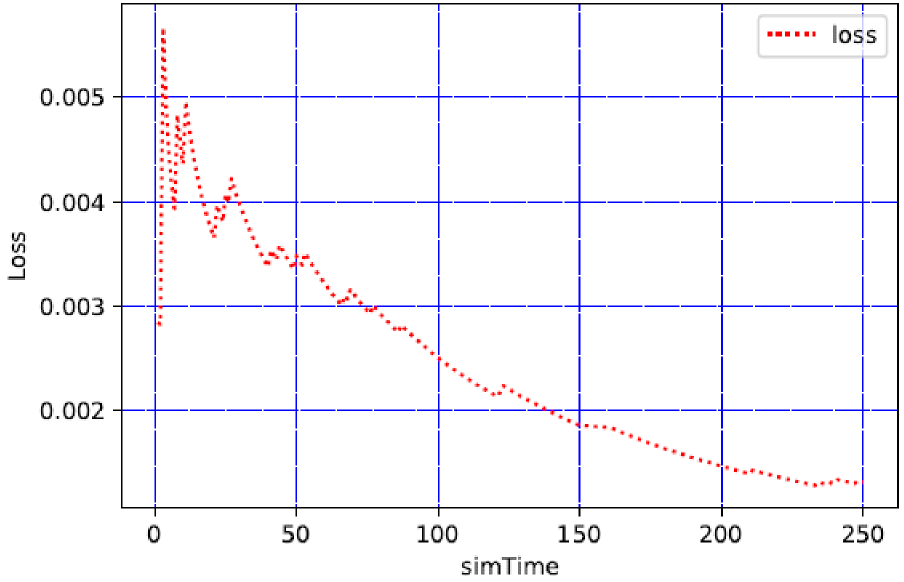
Performance of the DQN Algorithm
(Q Value, Loss, and Epsilon)

Q value



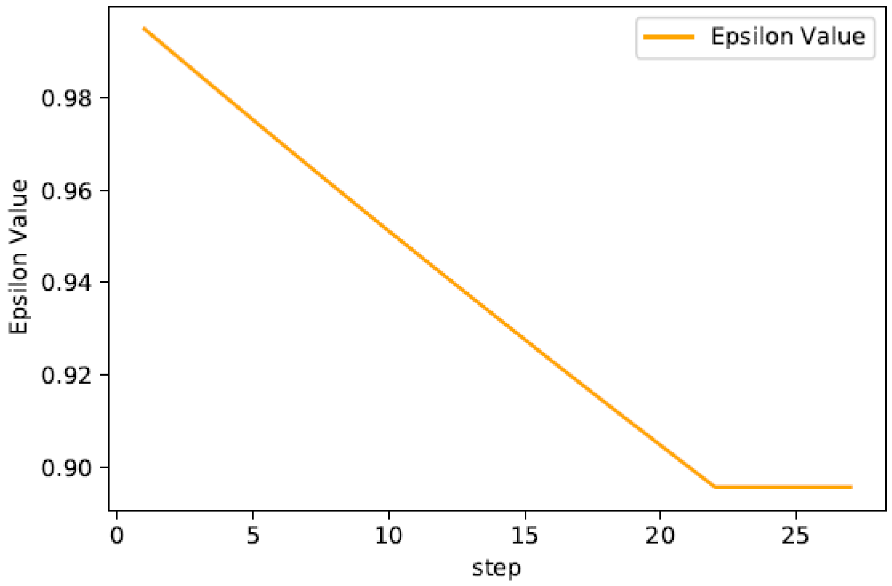
Plotting Cumulative Q value over episodes

LOSS



Loss values during training episodes

Epsilon Decay



Epsilon value for agent exploration

Detection Performance

Detection Performance

Model performance using typical multiagent IDS architecture

	Total cases	TP	TN	FP	FN	Accuracy	Precision
Classification	5651	624	682	2301	3350	0.19	0.21

Conclusion

- **An environment has been simulated for Reinforcement Learning based intrusion detection using simulation.**
- **Network performance measurement.**
- **Reinforcement Learning algorithm evaluation based on a single agent.**
- **Detection performance (accuracy, precision etc) is to be measured.**

Future Work

- **Other Reinforcement Learning algorithms can be tested using the framework.**
- **Cooperative Reinforcement Learning algorithms can also be tested and evaluated.**

Thank you

References

1. Jonsson, K. V. HttpTools: a toolkit for simulation of web hosts in OMNeT++. *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. Rome, Italy. 2009. 1–8.