Application of Learning Techniques to Resource Management in Virtual Networks

Rashid Mijumbi*, Juan-Luis Gorricho*, Joan Serrat*, Maxim Claeys†, Filip De Turck† and Steven Latré‡

*Universitat Politècnica de Catalunya, 08034 Barcelona, Spain

†Ghent University – iMinds, B-9050 Ghent, Belgium

‡University of Antwerp – iMinds, B-2020 Antwerp, Belgium

Tsinghua University, Beijing, PR China, December 17, 2013

Presentation Outline

Introduction

Ossification of the Internet

- The Internet, which evolved as an experimental packet switched network has grown beyond originally expected bounds.
- Resistance to changes at the Network Layer: Most changes are done at the end nodes, not routers.
  - Difficulties in IPv6 and IP Multicast Deployment

Ever Increasing User expectations

- Need for customisation of services and protocol Stacks through economically viable means
  - Separation between infrastructure and Service
  - Optimization of resource Usage and routing beyond BGP

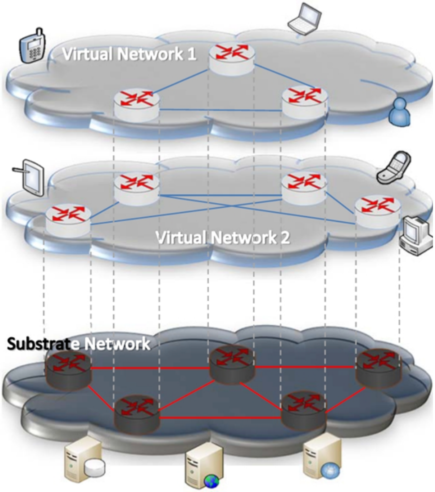
Network Virtualisation

Virtualisation

- Abstraction between user and physical resources.
- Potential strategy for addressing internet ossification.

Benefits

- Building, testing and experimentation of novel network architectures.
- Customised services and protocol stacks amoung SPs.



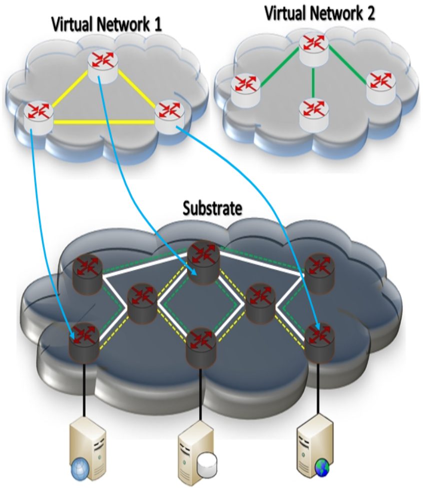
Problem: Efficient Resource Utilisation

Virtual Network Initialisation

- Resource Allocation not trivial
- Requires mapping virtual nodes and links to substrate nodes and links
- Computationally intractable

State-of-the-art

- Most current solutions static
- Dynamic ones allocate fixed resources



Distributed, Dynamic Resource Management (DDRM)

Observation

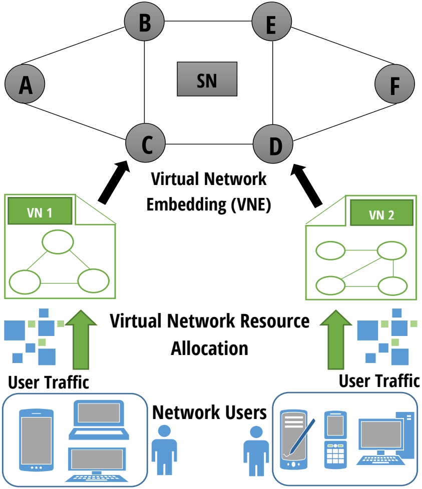
- Non-uniform internet traffic calls for better RM

2-Step Approach

- Virtual Network Embedding
- Distributed, Dynamic Resource Management

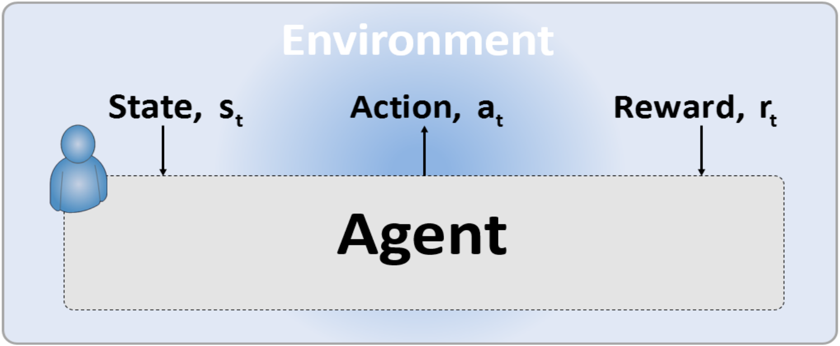
Proposed Method

- Artificial Intelligence, Reinforcement Learning, Multi-Agent Systems



Reiforcement Learning (RL)

- An agent is situated in an *environment*,
- Perceives *state*, s_t, and takes an *action*, a_t to change it,
- Receives a *reward*, r_t, which is an evaluation of its action
- Objective: maximise reward obtained in the long term



Q-Learning (I)

- Temporal difference algorithm used to determine the best actions to take in each possible state, by finding a *policy* that maximizes long term measure of reinforcement
- A *policy* defines the learning agent's way of behaving at a given time
- The action to be taken in a given state depends on the Q-values $Q(s, a)$ that are representative of the desirability of each action, a in that state, s.

Update Rule

$$Q(s_p, a_p) \leftarrow (1 - \alpha) Q(s_p, a_p) + \alpha \left\{ r_p + \lambda \max_{a \in \mathcal{A}} Q(s_n, a) \right\} \tag{1}$$

Q-Learning (II)

Action Selection

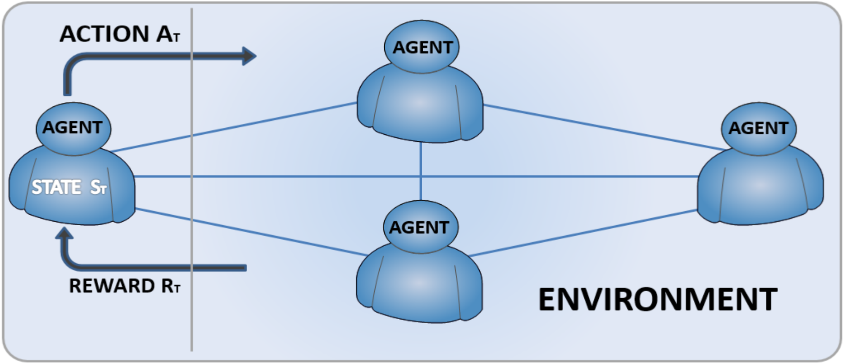
- ϵ-greedy: a greedy action is selected most of the time, and – using a small probability – a random action is chosen once in a while.
- Softmax: differs in how random action is selected. Assigns weights to all actions based on their values.

Softmax Action probability equation

$$\mathcal{P}(a|s) = \frac{\exp\{Q(s,a)/\tau\}}{\sum_{\hat{a}\neq a} \exp\{Q(s,\hat{a})/\tau\}} \tag{2}$$

Learning Multi-Agent Systems

- A group of agents located in the same environment
- Actions taken by a given agent could affect other agents
- Agents could be cooperative or competitive



Learning-based DDRM

Multi-agent System

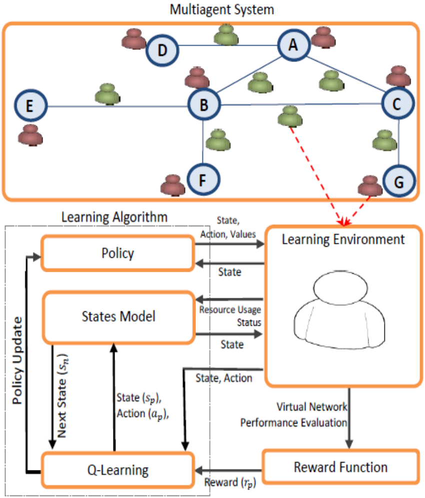
Distributed, an agent for each substrate node and link

Evaluative Feedback

Perceive, Act, Feedback - Reinforcement Learning

Q-Learning

Define: States, Actions, Values, Policy, Reward, Action Selection



Learning-based DDRM

Multi-agent System

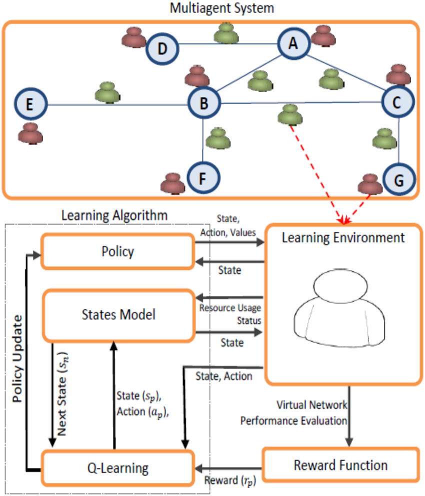
Distributed, an agent for each substrate node and link

Evaluative Feedback

Perceive, Act, Feedback - Reinforcement Learning

Q-Learning

Define: States, Actions, Values, Policy, Reward, Action Selection



Learning-based DDRM

Multi-agent System

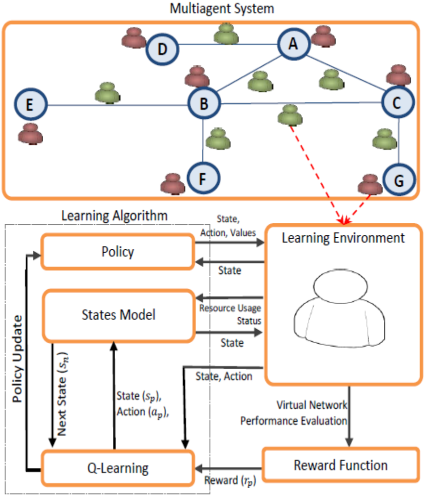
Distributed, an agent for each substrate node and link

Evaluative Feedback

Perceive, Act, Feedback - Reinforcement Learning

Q-Learning

Define: States, Actions, Values, Policy, Reward, Action Selection



Reinforcement Learning Model

States

- Vector **S** where each $s \in \mathcal{S}$ represents state of a virtual link/node. $s = \left(R_a,\ R^v_x,\ R^s_x\right)$
- Each state 9 bits, implying $2^9 = 512$ states

Actions

- Vector **A** where each $a \in \mathcal{A}$ represents action for virtual node/link

Code	Percentage Value
000	$0 < \text{Variable} \leq 12.5$
001	$12.5 < \text{Variable} \leq 25$
010	$25 < \text{Variable} \leq 37.5$
011	$37.5 < \text{Variable} \leq 50$
100	$50 < \text{Variable} \leq 67.5$
101	$67.5 < \text{Variable} \leq 75$
110	$75 < \text{Variable} \leq 87.5$
111	$87.5 < \text{Variable} \leq 100$

Action	Description
A_0	Maintain Currently allocated resources
A_1	Decrease allocated resources by 50.0 percent
A_2	Decrease allocated resources by 37.5 percent
A_3	Decrease allocated resources by 25.0 percent
A_4	Decrease allocated resources by 17.5 percent
A_5	Increase allocated resources by 17.5 percent
A_6	Increase allocated resources by 25.0 percent
A_7	Increase allocated resources by 37.5 percent
A_8	Increase allocated resources by 50.0 percent

Reward Function

$$r(v) = \begin{cases} -100 & \text{if } R_a \leq 0.25 \\ \nu R_u - \left(\kappa \hat{D}_{ij} + \eta \hat{P}_i\right) & \textit{otherwise} \end{cases}$$

- Objective is to encourage high virtual resource utilisation while punishing $n_a \in \mathcal{N}_a$ for dropping packets and $l_a \in \mathcal{L}_a$ for having a high delay
- Punitive reward of -100 to resource allocations below 25% to ensure that this is the minimum allocation to a virtual resource.

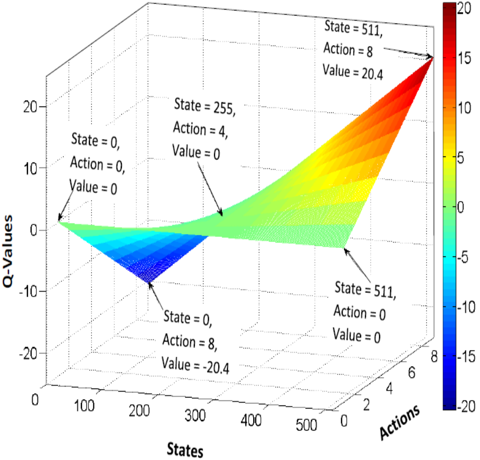
Policy

Implemented by lookup table. 9 possible actions, 512 possible state $\Rightarrow$ Policy size is $9 \times 512 = 4608$

Initialisation

- Q-learning requires all state-action pairs to be visited at least once so as to reach optimality
- A random or constant initial values may lead to a slow convergence especially for a policy with many state-action values like we have in our approach

$$Q(s,a) = \frac{a}{\Psi} \times (s - 255) \quad (3)$$



Learning Algorithm

Action Selection

Evaluation of softmax and ϵ-greedy

Agent Cooperation

Conflict avoidance for substrate link agents

Algorithm 1 *Agent Learning Algorithm*

POLICY INITIALISATION:
for $s \in \mathcal{S}, a \in \mathcal{A}$ **do**
  Initialize the Q-table values $Q(s, a)$
end for
Determine current state $s(c)$
previous state, $s_p = s_c$, previous action, $a_p = A_0$, next state, s_n.
repeat
  Wait(Learning Interval)
  POLICY UPDATE:
  Read s_p, a_p, s_n
  Observe Virtual Network Performance and Determine reward for previous action r_p.
  Update the Q-Table using the equation (1)
  ACTION SELECTION:
  Determine current state, s_c.
  Choose an action, $a_c \in \mathcal{A}$, for that state using a given *action selection criterion*
  Take the action, a_c and determine next state s.
  Set $s_p = s_c$, $a_p = a_c$, $s_n = s$
until Learning is stopped

Simulation Setup

NS3 Parameters

Parameter	Value
Queue Type	Drop Tail
Queue drop Mode	Bytes
Maximum Queue Size	6,553,500 Bytes
Maximum Packets Per VN	3500 Packets
Number of VNs	1024
Network Mask	255.255.224.0
IP Adress Range	10.0.0.0 – 10.255.224.0
Network Protocol	IPv4
Transport Protocol	TCP
Packet MTU	1518 Bytes
Packet Error Rate	0.000001 per Byte
Error distribution	Uniform (0, 1)
Port	8080

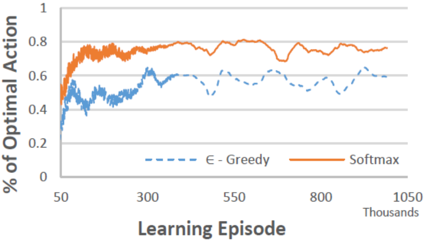
Brite Paremeters

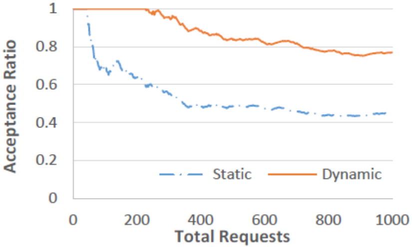
Parameter	Substrate Network	Virtual Network
Name (Model)	Router Waxman	Router Waxman
Number of nodes (N)	25	[5 - 10]
Size of main plane (HS)	250	250
Size of inner plane (LS)	250	250
Node Placement	Random	Random
GrowthType	Incremental	Incremental
Neighbouring Nodes	3	2
alpha (Waxman Parameter)	0.15	0.15
beta (Waxman Parameter)	0.2	0.2
BWDist	Uniform	Uniform
Minimum BW (BWMin)	2×10^6 bps	1×10^6 bps
Maximum Dev. (BWMax)	8×10^6 bps	1×10^6 bps

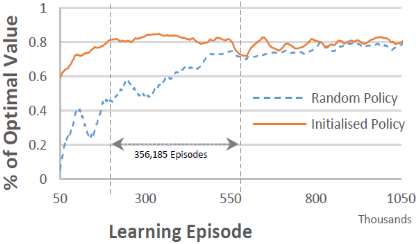
Parameter	Substrate Network	Virtual Network
Minimum Number of Nodes	25	5
Maximum Number of Nodes	25	10
Minimum Node Queue Size	(100×1518) Bytes	(10×1518) Bytes
Maximum Node Queue Size	(200×1518) Bytes	(20×1518) Bytes
Minimum Link Bandwidth	2.0Mbps	1.0Mbps
Maximum Link Bandwidth	10.0Mbps	2.0Mbps

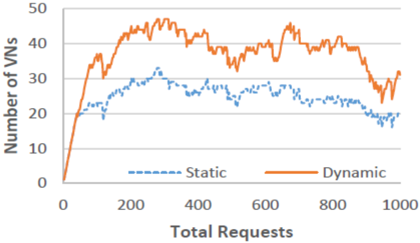
Substrate and Virtual Network Properties

Evaluations I

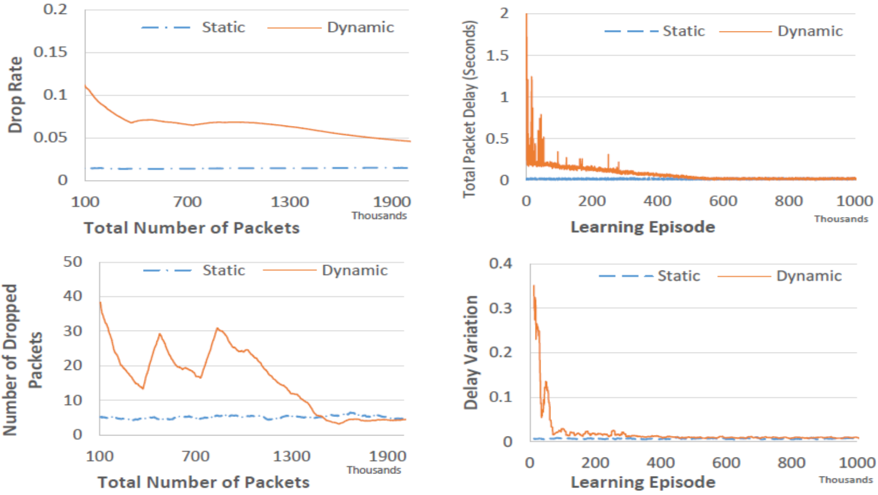








Evaluations II



Conclusion

- DDRM improves resource utlisation efficiency, better revenue for InPs
- When agents have learnt optimal policies, QoS is comparable to static approach
- Initialising Learning process enhances speed of convergence to optimal actions
- Softmax action selection method is best suited for this task

References

[1] Rashid Mijumbi, Juan-Luis Gorricho, Joan Serrat, Maxim Claeys, Steven Latre, Filip De Turck, "Design and Evaluation of Learning Algorithms for Dynamic Resource Management in Virtual Networks", IEEE/IFIP Networks Operations and Management Symposium, (NOMS), Krakow, Poland. (2014)

[2] Rashid Mijumbi, Joan Serrat and Juan Luis Gorricho, "Learning Algorithms for Dynamic Resource Allocation in Virtualised Networks", Management of Large Scale Virtualized Infrastructures: Smart Data Acquisition, Analysis and Network and Service Management in the Future Internet. (2014)

[3] Rashid Mijumbi, Juan-Luis Gorricho, Joan Serrat, "Contributions to Efficient Resource Management in Virtual Networks" 8th International Conference on Autonomous Infrastructure, Management and Security (AIMS), Brno, Czech Republic. (2014)

[4] Rashid Mijumbi, Joan Serrat and Juan Luis Gorricho, "Multi-Agent Based Resource Allocation in Next Generation Virtual Networks" Proceedings of the 2012 Barcelona Forum on Ph.D. Research in Communication and Information Technologies, Barcelona, Spain, (2012)

[5] Rashid Mijumbi, Juan-Luis Gorricho, Joan Serrat, Javier Rubio-Loyola, Ramon Aguero, "Survivability-oriented Negotiation Algorithms for Multi-domain Virtual Networks", 10th International Conference on Network and Service Management (CNSM 2014), Rio de Janeiro, Brazil.

[6] Rashid Mijumbi, Joan Serrat and Juan Luis Gorricho, "Autonomic Resource Management in Virtual Network", Scalable and Adaptive Internet Solutions (SAIL) Summer School Work in Progress Session, Santander, Cantabria, June 2012

[7] Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, "Self-managed Resources in Network Virtualization", IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa Canada, May 2015. Dissertation Digest Paper

[8] Rashid Mijumbi, Joan Serrat, Javier Rubio-Loyola, Niels Bouten, Steven Latre, Filip De Turck, "Dynamic Resource Management in SDN-based Virtualized Networks", IEEE/IFIP CNSM, International Workshop on Management of SDN and NFV Systems, Rio de Janeiro, Brazil

[9] Rashid Mijumbi, Juan-Luis Gorricho, Joan Serrat, Maxim Claeys, Jeroen Famaey, Filip De Turck, "Neural Network-based Autonomous Allocation of Resources in Virtual Networks", European Conference on Networks and Communications (EuCNC), 2014, Bologna, Italy, June 2014

Thank You

THANK YOU!!!

Contact: rashid@tsc.upc.edu