Cognitive Packet Network

A dissertation submitted in the partial fulfillment for the award of Degree of

Master of Technology

In

Software Technology

by

Kishore Chandra Dixit (Roll no. 2K11/ST/10)

Under the guidance of

Vinod Kumar

DEPARTMENT OF COMPUTER ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

BAWANA ROAD, DELHI

2014

DECLARATION

I hereby want to declare that the thesis entitled "**Cognitive Packet Network**" which is being submitted to the **Delhi Technological University**, in partial fulfillment of the requirements for the award of degree in **Master of Technology in Software Technology** is an authentic work carried out by me. The material contained in this thesis has not been submitted to any institution or university for the award of any degree.

Kishore Chandra Dixit

Department of Computer Engineering

Delhi Technological University,

Delhi-110042

CERTIFICATE

DELHI TECHNOLOGICAL UNIVERSITY

BAWANA ROAD, DELHI-110042

Date: ____________________

This is to certify that the thesis entitled "**Cognitive Packet Network** "submitted by **Kishore Chandra Dixit (Roll Number: 2K11/ST/10),** in partial fulfillment of the requirements for the award of degree of Master of Technology in Software Technology, is an authentic work carried out by him under my guidance. The content embodied in this thesis has not been submitted by him earlier to any institution or organization for any degree or diploma to the best of my knowledge and belief.

Vinod Kumar,

Assoc. Professor,

Department of Computer Engineering,

Delhi Technological University, Delhi-110042

ACKNOWLEDGEMENT

I would like to take this opportunity to express my appreciation and gratitude to all those who have helped me directly or indirectly towards the successful completion of this work.

Firstly, I would like to express my sincere gratitude to my guide **Mr.Vinod Kumar**, **Associate Professor, Department of Computer Engineering**, **Delhi Technological University, Delhi** whose benevolent guidance, encouragement, constant support and valuable inputs were always there for me throughout the course of my work. Without his continuous support and interest, this thesis would not have been the same as presented here.

Besides my guide, I would like to thank **Mr. Kamal Garg**, M Tech. Scholar DTU for his help. Also I would like to extend my thanks to the entire staff in the Department of Software Engineering, DTU for their help during my course of work.

KISHORE CHANDRA DIXIT

2K11/ST/10

ABSTRACT

The development of computer networks has seen a paradigm shift from static, hierarchical network structures to highly distributed, autonomous systems without any form of centralized control. For networking nodes the ability to self-adapt and self-organize in a changing environment has become a key issue. One of the main challenges faced by computer networks is the efficient management of increasing complexity.

Here we discuss the definition and framework for a novel type of adaptive data network: **The cognitive network**. In a cognitive network, the collection of elements that make up the network observes network conditions and then, using prior knowledge gained from previous interactions with the network, plans, decides and acts on this information. They are capable of perceiving current network conditions and then planning, learning, and acting according to end-to-end goals. Cognitive networks are motivated by the complexity, heterogeneity, and reliability requirements of tomorrow's networks, which are increasingly expected to self-organize to meet user and application objectives.

In Cognitive Packets Network (CPN) the intelligent capabilities for routing and flow control are concentrated in the packets, rather than in the nodes and protocols. Cognitive packets within a

CPN route themselves. They are assigned goals before entering the network and pursue these goals adaptively. Cognitive packets learn from their own observations about the network and from the experience of other packets with which they exchange information via mailboxes. Cognitive packets rely minimally on routers.

Here we have tried to present the advantages of CPN over the legacy IP routing protocol with the help of various measurement which we have performed under identical conditions showing the gain resulting from the use of CPN.

Table of Contents

List of Figures

CHAPTER 1

INTRODUCTION

Today our reliance on computer networks is a fact of life; as such the adaptability, security and reliability of these networks are of utmost importance. Increased connectivity and availability of the world's networks has lead to more and more complex systems, with an equally increasing number of vulnerabilities.

Reliable networks that provide good service quality are expected to become the norm in all communication aspects, especially as the information transferred between network users gets more complex and confidential, and as malicious users try to deliberately degrade or altogether deny legitimate network service. There is therefore an increased need for network adaptability, stability and reliability which has led to the growth of autonomic networks that use QOS-driven approaches for greater stability and reliability in communications.

Here we will discuss about a packet switching networks in which intelligence is constructed into the packets, rather than at the nodes or in the protocols. Networks which contain such packets will be called "**Cognitive Packet Networks (CPN)".** Cognitive packets in CPN route themselves, and learn to avoid congestion and to avoid being lost or destroyed.

Cognitive packets learn from their own observations about the network and from the experience of other packets. They rely minimally on routers. Each cognitive packet progressively updates its own model of the network as it travels through the network, and uses the model to make routing decisions.

In the most extreme case, a cognitive packet will "know" where it is in the network without asking for the identity of the switch where it is being currently stored, so that packets can be self-routed without relying on the routing algorithms provided by the network nodes. Cognitive packets rely minimally on routers, so that network nodes only serve as buffers, mailboxes and processors.

In CPN, the steady-state of Random Neural Networks (RNN) [12] running on network routers decides network paths. Random Neural Networks (RNNs) are trained continuously by enhancing network flows with packets that monitor performance. These packets store information about the network state as they cross the network from a source to a destination, and get updated at each hop in the path. The Cognitive Packet Network provides good network adaptation properties under varying network conditions and user requirements. They can perceive current network conditions, plan, decide, act on those conditions, learn from the consequences of its actions, all while following end-to-end goals.

Cognitive Packet Network

The Cognitive Packet Network (CPN) is a routing protocol that uses adaptive techniques based on on-line measurements to provide QoS to its users. The users themselves can declare individually their QoS goals, such as minimum delay, minimum packet loss, maximum bandwidth, minimum power consumption or a weighted combination of these. CPN has been designed to perform self-improvement in a distributed manner by learning from the experience of the packets in the network and by constantly probing for the current best routes.

Cognitive packets in CPN route themselves, and learn to avoid congestion and to avoid being lost or destroyed. Cognitive packets learn from their own observations about the network and from the experience of other packets with which they exchange information via mailboxes. Each cognitive packet progressively refines its own model of the network as it travels through the network, and uses the model to make routing decisions. In the most extreme case, a cognitive packet will know, whether it is in the network without asking for the identity of the node where it is being currently stored. So the packets can be self-routed without relying on the routing algorithms provided by the network nodes.

1.1 Structure of Cognitive Packets

A Cognitive Packet (CP) contains the following fields:

- The Identifier Field (IF) which provides a unique identifier for the CP, as well as information about the class of packets it may belong to, such as its quality of service (QoS) requirements.
- The Data Field contains the data/payload. It contains the real information which has to be transferred from source to destination.
- A Cognitive Map (CM) which contains the usual Source and Destination (S-D) information, as well as a map showing where the packet currently .thinks. it is, the packet's view of the state of the network, and information about where it wants to go next.
  The S-D information may also be stored in the IF.
- Executable code that the CP uses to update its CM. This code will contain learning algorithms for updating the CM, and decision algorithms which use the CM.

Identifier Field	DATA	Cognitive Map	Executable Code

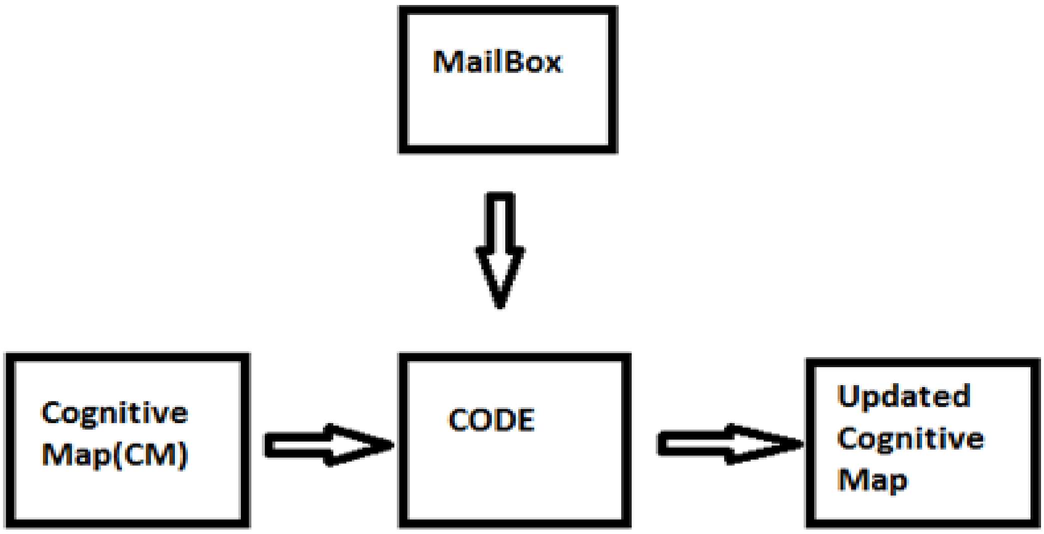
<u>Fig-1 Structure of Cognitive Packets.</u>

CPs store information in their private Cognitive Map (CM) [20] and update the CM and make their routing decisions using the code which is in each packet. This code will include neural networks or other adaptive algorithms which will be described above.

In a CPN, the packets use nodes act as .parking. Or resting areas where they make decisions and route themselves. They also use nodes as places where they can read their mailboxes.

Mailboxes may be filled by the node, or by other packets which pass through the node. Packets also use nodes as processors which execute their code to update their CM and then execute their routing decisions. As a result of code execution, certain information may be moved from the CP to certain mailboxes. The nodes may execute the code of CPs in some order of priority between classes of CPs, for instance as a function of QoS requirements which are contained in the identification field).

The manner in which Cognitive Memory at a Node is updated by the node's processor is shown in Figure 2.



<u>Fig-2 Update of a CP by a node CPN</u>

Packet arrives at the input buffer of a node. The code of each CP in the input buffer is executed and the following is done.

- The CP retrieves relevant information from the mailbox
- The packet's CM is updated
- Some information is moved from the CP to the MB
- The CP is moved to an output buffer

A node in the CPN acts as a storage area for CPs and for mailboxes which are used to exchange data between CPs, and between CPs and the node.

It has an input buffer for CPs arriving from the input links, a set of mailboxes, and a set of output buffers which are associated with output links.

Nodes in a CPN carry out the following functions:

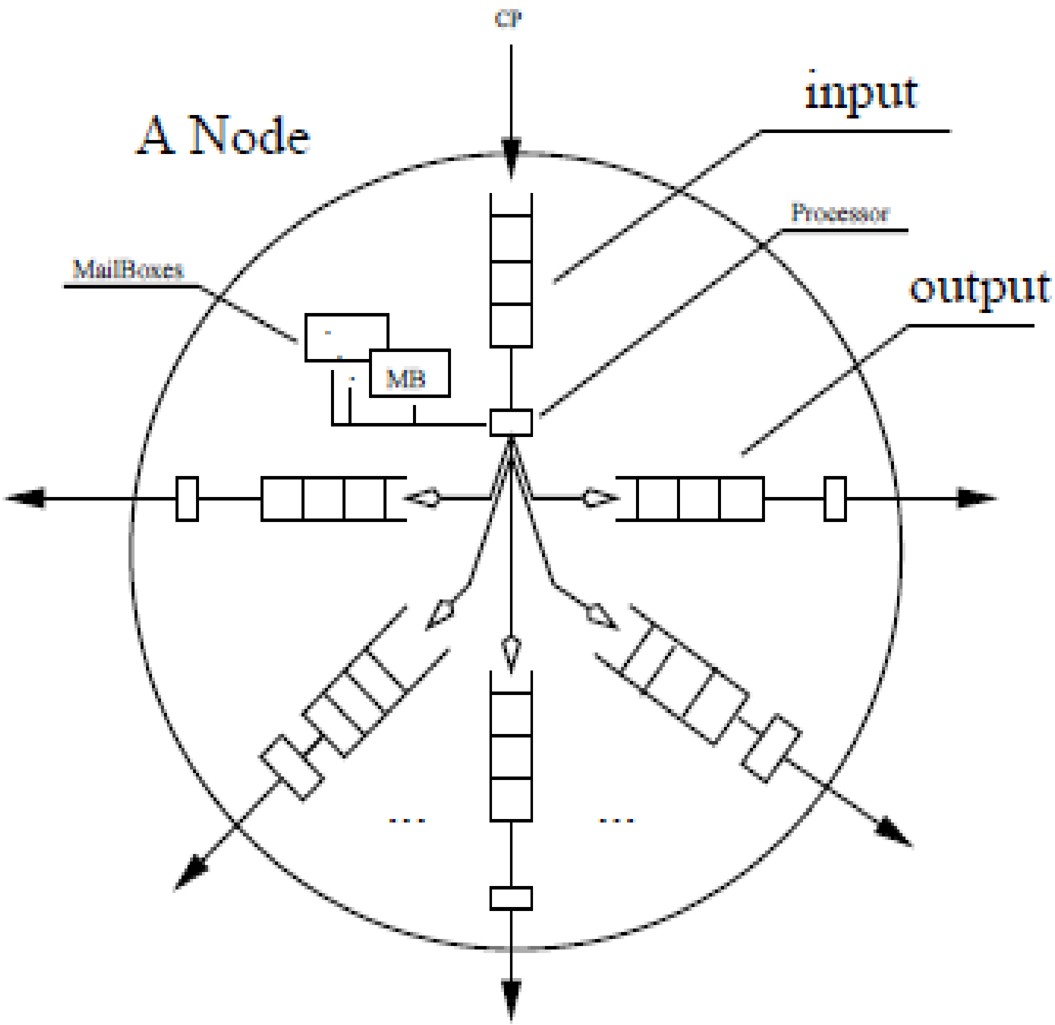
1. A node receives packets via a finite set of ports and stores them in an input buffer.
2. It transmits packets to other nodes via a set of output buffers. Once a CP is placed in an output buffer. It is transmitted to another destination node with some priority indicated in the output buffer.
3. A node receives information from CP,s which it stores in Mailboxes(MB).MB,s may be reserved for certain classes of CP,s or may be specialized by classes of CP,s. For instance, there may be different MB,s for packets identified by different Source-Destination pairs.
4. A node executes the code for each CP in the input buffer [17]. During the execution of the CPs code, the CP may ask the node to decline its identity, and to provide information about its local connectivity (i.e. “This is Node A, and I am connected to Nodes B, C, D via out-put buffers) while executing its code. In some cases, the CP may already have this information in its CM as a result of the initial information it received at its source, and as a result of its own memory of the sequence of moves it has made.

Cognitive packet routing is carried out using a reinforcement learning (RL) algorithm based on Random Neural Networks (RNN). The algorithm code is stored in each router and its parameters are updated by the router. For each successive smart packet, the router computes the appropriate outgoing link based on the outcome of this computation.

So we can conclude the below:-

- Storage area for CPs: Input and Output Buffers
- Mailboxes are used to exchange data between CPs
- The Node executes the code for each CP in the input buffer.

A recurrent RNN, with as many “neurons” as there are possible outgoing links, is used in the computation of CPN. The weights of the RNN are updated so that decision outcomes are reinforced or weakened depending on how they have contributed to the success of the QoS goal.



<u>Fig-3 Node in a CPN</u>

As a result of this execution in a node:

- The CM's of the packets in the input buffer are updated,
- Certain information is moved from CPs to certain MB's,
- A CP which has made the decision to be moved to an output buffer is transferred there, with the priority it may have requested.

CPN makes use of adaptive techniques to seek out routes based on user defined QoS criteria. For in- stance, packet loss and delay can be used as routing criteria to improve overall reliability for the users of the network, or delay and its variance can be used to find routes which provide the QoS requested by voice packets.

1.2 Types of CPN Packets

A CPN carries three types of packets:-

- Smart packets (SP)
- Dumb packets (DP)
- Acknowledgments (ACK).

Smart or cognitive packets route themselves, they learn to avoid link and node failures and congestion and to avoid being lost. They learn from their own observations about the network and from the experience of other packets. They rely minimally on routers. Smart packets use reinforcement learning to discover routes, and the reinforcement learning “re-ward” function incorporates the QoS requested by a particular user. This reward is the inverse of a QoS “goal” which each user can provide before initiating a connection.

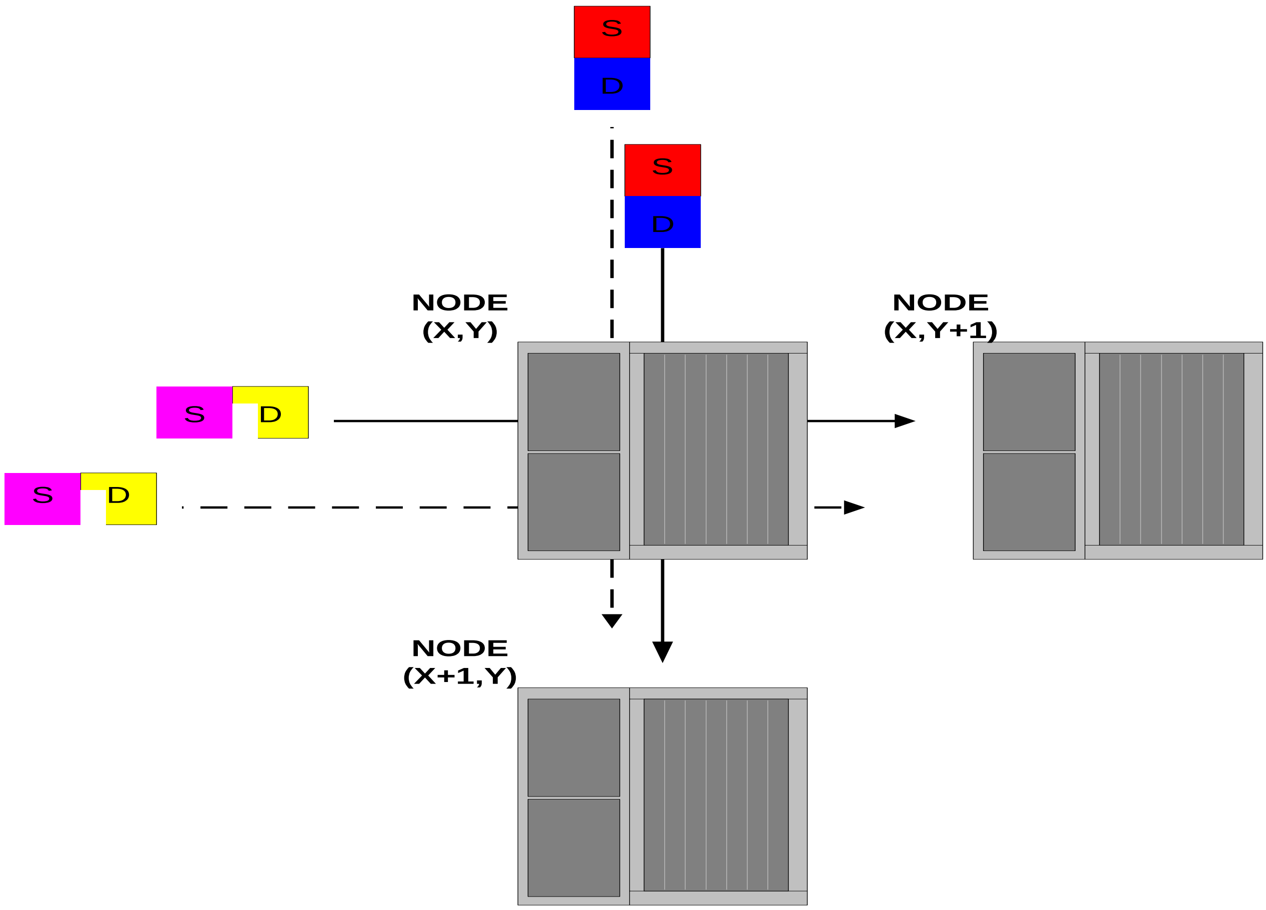
When a smart packet arrives to a destination, an acknowledgment (ACK) packet is generated by the destination and the ACK heads back to the source of the smart packet along the inverse route. As it traverses successive routers, it updates mailboxes in the CPN routers;

When it reaches the source node it provides source routing information for the dumb packets. Dumb packets of a specific QoS class use successful routes which have been selected in this manner by the smart packets of the same class.

1.3 Classes of CPN packets

Cognitive packets can be hierarchically arranged into various classes [5],[8].
A Class is a set of packets having the same QoS requirements, sets of internal states, control rules, input-output signals.

The nodes of a Cognitive packet network may handle various cognitive packets based on their class they may assign a set of mailbox for a particular class of cognitive packet.
The routing information carried by ACK packets of a specific class is passed to dumb packets of same class only.



<u>Fig-4 Classes of Cognitive Packets</u>

1.4 STD of a CPN Router

The below diagram shows the various states of a CPN router which handles the CPN packets. All these activities are carried by the CPN router after the packet has arrived in the input buffer. The packets are send to output buffer as per the below diagram.

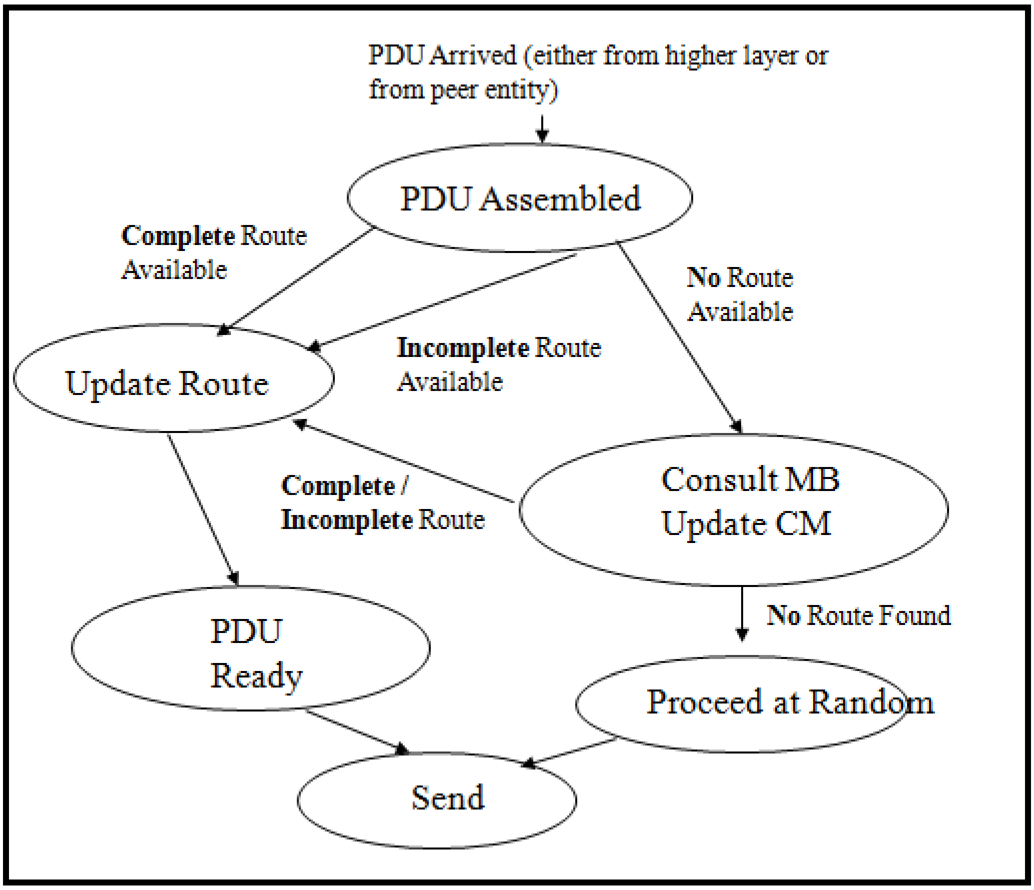


Fig 5 State transition diagram or a CPN Router

CHAPTER 2

2.1 Adaptation by cognitive packets

Each cognitive packet starts with a given representation of the network from which it then progressively constructs its own Cognitive Map of network state and uses it to make routing decisions. Learning paradigms are used by CPs to update their CM and reach decisions using the packet's prior experience [12],[15] and the input provided via mailboxes.

In the adaptive approach we propose for CPs, each packet entering the network is assigned a Goal before it enters the net-work, and the CP uses the goal to determine its course of action each time it has to make a decision. For instance if the CP contains part of a telephone conversation, a typical goal assigned to the packet might be: “Go from source S to destination D in minimum time”. A more sophisticated goal in this case could be: “go from S to D in minimum time, but do not overtake any packets of the same sequence which left before you”, since voice packets need to be used at the receiver in the sequence they were transmitted. On the other hand, if these were data packets, the goal may simply be: “go from S to D without getting lost or destroyed”.

In our work, these goals are translated into numerical quantities (e.g. delay values, loss probabilities, and weighted combinations of such numerical quantities) which are then used directly in the adaptation. A simple approach to adaptation is to respond in the sense of the most recently available data. Here the CPs cognitive memory contains data which is up-dated from the contents of the node's mailbox. After this update is made, the CP makes the decision which is most advantageous (lowest cost or highest reward) simply based on this information; we will call this approach the Bang-Bang algorithm.

We will be using the below learning paradigms for CPs.

Random neural networks with reinforcement learning (RNNRL) [16].

In this case a recurrent network is used both for storing the CM and making decisions. The weights of the network are updated so that decisions are reinforced or weakened depending on whether they have been observed to contribute to increasing or decreasing the accomplishment of the declared goal. A description of the RNN and the related algorithms used in this paper are presented in next section.

2.2 The Random Neural Network and related algorithms

For the Reinforcement learning (RL)[12] approach to CP adaptation, as well the feed-forward neural network predictor [18], we have used the RNN, which is an analytically tractable spiked random neural network model whose mathematical structure is akin to that of queuing networks.

The state $\mathbf{q_i}$ of the i-th neuron in the network represents the probability that i-th neuron is excited. Each neuron is associated with a distinct outgoing link at a node.
The state $\mathbf{q_i}$ should satisfy the following system of non-linear equations:-

$$q_i = \frac{\lambda_i^+}{r_i + \lambda_i^-}$$

Where

$$\lambda_i^+ = \sum_j q_j w_{ji}^+ + \Delta_i$$

$$\lambda_i^- = \sum_j q_j w_{ji}^- + \lambda_i$$

W_{ij}^+ is the rate at which neuron i sends excitation spikes to neuron j when neuron i is excited.

W_{ij}^- is the rate at which neuron i sends inhibition spikes to neuron j when i is excited

and r(i) is the total firing rate from the neuron i.

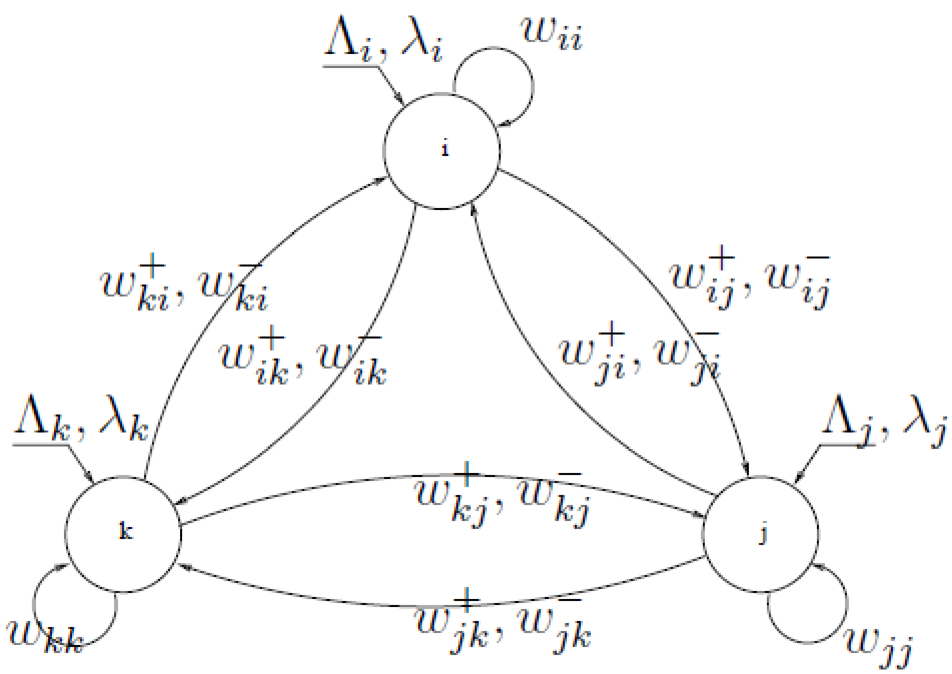


Fig-6 Example of RNN

Where $w^+_{ij} = r(i)p^+(i,j)$ and $w^-_{ij} = r(i)p^-(i,j)$

Are the rates at which neuron *i* sends excitation and inhibition spikes to neuron j respectively.

For a neuron network, the network parameters are these n by n "weight matrices", which need to be "learned" from input data.

RL is used in CPN as follows. Each node stores a specific RNN for each active source-destination pair, and each QoS class. The numbers of nodes of the RNN are specific to the router, since (as indicated earlier) each RNN node will represent the decision to choose a given output link for a smart packet.

Decisions are taken by selecting the output link j for which the corresponding neuron is the most excited, i.e. $q_i < q_j$ for all i=1,2….n.

Each QoS class for each source-destination pair has a QoS goal **G**, which expresses a function to be minimized, e.g., Transit Delay or Probability of Loss, or Jitter, or a weighted combination, and so on.
The reward **R** which is used in the RL algorithm is simply the inverse of the goal

$$R = G^{-1}$$

Successive measured values of R are denoted by R_l, where l=1, 2…

These are first used to compute the current value of the decision threshold **T**:

$$\mathbf{T_l = \beta T_{l-1} + (1 - \beta)R_l \ ; \ 0 < \beta < 1}$$

Where **β** is some constant 0<**β**<1, typically close to 1.

Suppose we have now taken the *l* th decision which corresponds to neuron *j* and that we have measured the *l* th reward $\mathbf{R_l}$. We first determine whether the most recent value of the reward is larger than the previous value of the threshold $\mathbf{T_{l-1}}$.

If that is the case, then we increase very significantly the excitatory weights going into the neuron that was the previous winner (in order to reward it for its new success), and make a small increase of the inhibitory weights leading to other neurons.

If the new reward is not greater than the previous threshold), then we simply increase moderately all excitatory weights leading to all neurons, except for the previous winner, and increase significantly the inhibitory weights leading to the previous winning neuron (in order to punish it for not being very successful this time).

Learning comes from past experience:

If $T_{l+1} \leq R_{l+1}$

$w_{ji}+ \leftarrow w_{ji}^{+} + R_l$

$w_{ji}^{-} \leftarrow w_{ji}^{-} + nR{-}l_2$, $k = j$, **(Reward)**

Else $w_{ji}+ \leftarrow w_{ji}+ + nR{-}l_2$, $k = j$,

$w_{ji}^{-} \leftarrow w_{ji}^{-} + R_l$. ***(Punishment)***

$\mathbf{R}_l$ is calculated from information stored in Mailbox.

Since the relative sizes of the weights of the RNN, rather than the actual values, determine the state of the neural network, we then re-normalize all the weights by carrying out the following operations.

Normalization

First for each *i* we compute:

$$r_i^* = \sum_1^n [w^+(i,m) + w^-(i,m)]$$

And then re-normalize the weights with:

$$w^+(i, j) \leftarrow w^+(i, j) + \frac{r_i}{r_i^*}$$

$$w^-(i, j) \leftarrow w^-(i, j) + \frac{r_i}{r_i^*}$$

Finally, the probabilities qi are computed using the nonlinear iterations. The largest of the qi's is again chosen to select the new output link used to send the smart packet forward. This procedure is repeated for each smart packet for each QoS class and each source-destination pair.

CHAPTER 3

3.1 Simulation of Cognitive packet network

The purpose of our simulation was to see whether algorithms which can be implemented inside the CPs can result in delay and loss performance improvements as compared to normal IP networks.

Under identical traffic conditions. A single network simulation program was written, and three different learning algorithms were used by the CPs. The CPN was chosen to be quite large with 20 nodes, and we chose a locally interconnected rectangular grid topology as shown in Figure 10.All link speeds were normalized to 1, and packets were allowed to enter and leave the network either from the top ten nodes or the bottom ten nodes.

Traffic arrival into the network was taken to be Poisson. Each packet's destination was one of the nodes at the opposite end of the network, and the destination node for each packet were chosen to be fixed when the packet enters the network, and drawn to be equally likely among all possible 10 destination nodes.

Buffers in each node are of unlimited capacity so that blocking or loss is not tied to congestion. Packet loss was simulated probabilistically at all nodes with a small fixed probability of loss throughout all but a few specific nodes where there is a high packet loss probability; these latter nodes are not known in advance by the packets.

Congestion at a node can be caused by the normal packet traffic and routing, or by packets which route to certain nodes because of congestion elsewhere in the network. It can also occur when packets remain in a “safe” node due to the risk of congestion or loss in other parts of the network. All the packets were assigned a common goal, which was to minimize a weighted combination of delay (W) and loss (L).

$$\mathbf{G = W + L}$$

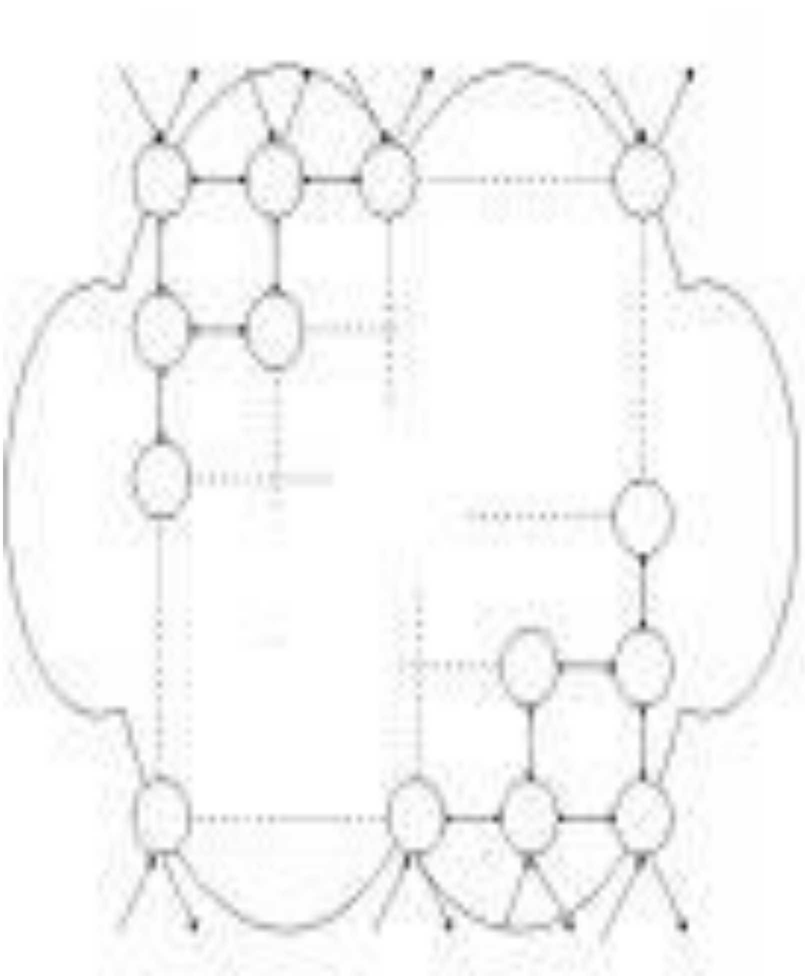


Fig-7 The Simulated CPN Topology

All the algorithms uses four items of information which are deposited in the nodes' mailboxes: (1)The length of the local queues in the node,(2) recent values of the downstream delays experienced by packets which have previously gone through the output links and reached their destinations, (3) the loss rate of packets which have passed through the same node and gone through the output links, (4) estimates made by the most recent CPs which have used the output links headed for some destination d of its estimated delay D_d and loss L_d from this node to its destination.

The value D_d is updated by each successive CP passing through the node and whose destination is *d* as discussed below.

Only a fraction of the packets to any destination are marked for monitoring. The departure time-stamps of these marked packets are stored in each node's mailboxes, and the arrival dates to destination for the same marked packets arrive via acknowledgement (ACK) packets sent back by the destination nodes. These two times used to reconstruct the downstream delays for each node.

In the Bang-Bang algorithm, the CPs read the mailboxes and for each destination compute an estimated running average delay of the form W_d aWd + (1 , a)V_d where V_d is the most recently available downstream delay value to destination *d* based on the current decision. Similar information is collected and updated for packet loss l_d.

The CP then makes an assessment of what a “reasonable” value of downstream Loss L_d and Delay D_d should be to its destination d, based on its knowledge of where it currently is in the network, of its distance to the destination node, and using the output queue lengths at the node.

If either $L_d < l_d$ or $D_d < W_d$, then the CP selects at random any other output link which is on some path to the destination *d*.

Throughout the simulation results, we vary the arrival rates of packets to each input node between 0:1 and 1. All simulations compare the CPs with controls of different kinds, Normal IP traffic which has a static routing policy where the packet is routed along a static shortest path to the output layer of nodes, and then horizontally to its destination.

Clearly, this static algorithm could also be implemented in a CPN. However the adaptive algorithms we have simulated allow each individual CP to carry out a separate computation and make its own individual decisions; thus these adaptive algorithms are a better illustration of the capabilities which can be implemented in a CPN as compared to our legacy networks using IP.

Simulations results which compare the Cognitive approach to normal IP networks shortest-path type routing are shown in Fig 9, 10, 11.

The simulation results obtained in next section will show that the average delay is reduced by 10% in the CPN as compared to normal IP network where the packets have no intelligence and are dependent on the nodes for their routing. Also the packet loss is decreased significantly by almost 70 to 80 % in CPN.

3.2 The normal CPN implementation

We have tried to capture the path that a normal IP network will follow and the path which may be selected by cognitive packets. The path selected by cognitive packets is not constant it varies depending upon the various conditions like traffic, delay and packet loss.The below figure shows the basic CPN implementation where

– Pink dots with blue boundary are nodes.
– Blue dotted line is link between various nodes.
– Bold red line is route formed by our cognitive packet algorithm.
– Bold green line is normal route formed on the basis of distance.

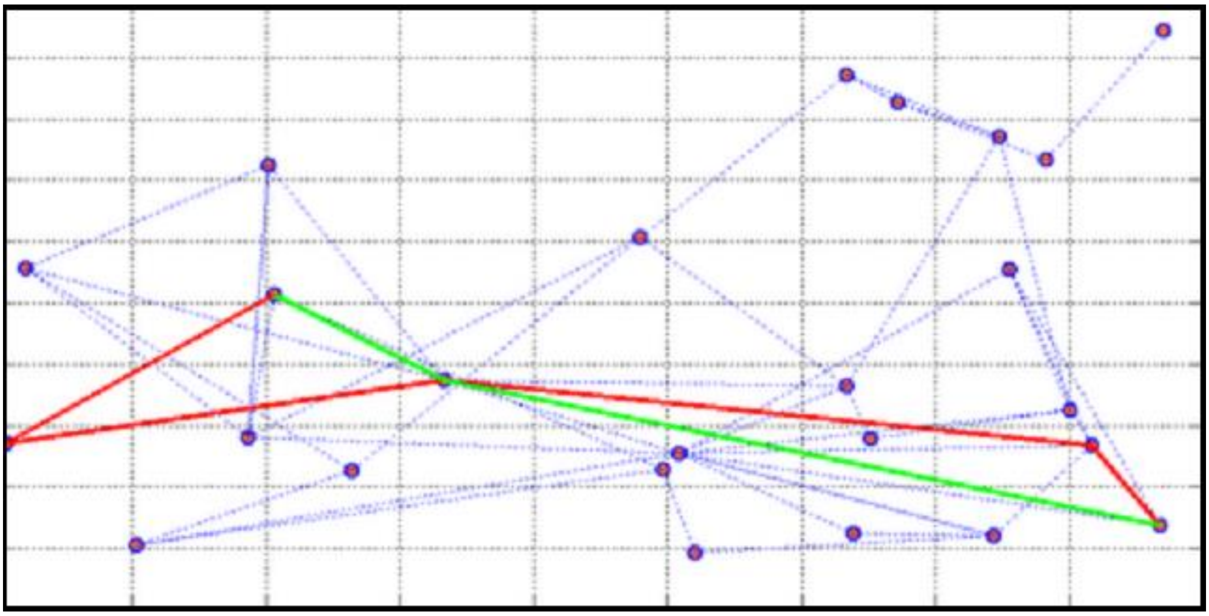


Fig-8 The CPN implementation

As can been seen the above figure,the cognitive packets finds the route based on the intelligence.This route must be the most efficient route which has been chosen by the smart packets considering all the various factors.

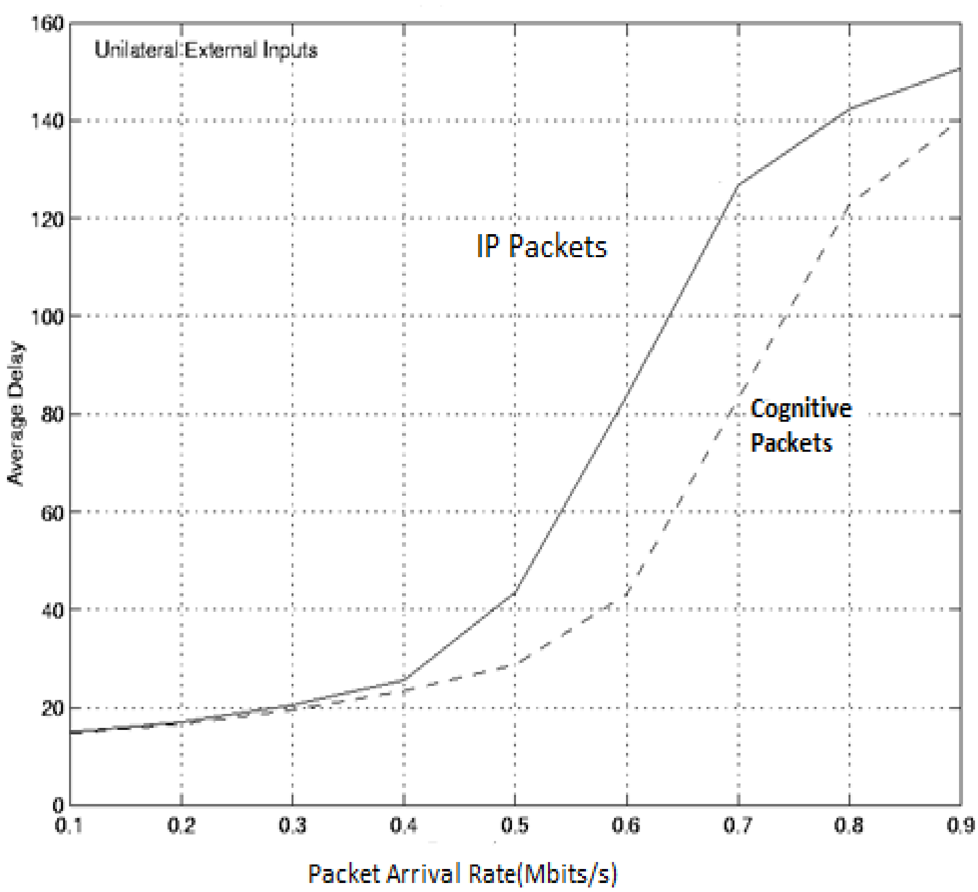
3.3 Comparision of Average Delay in CPN and IP Network

In CPN the dumb packets carry the payload or the actual information.So their performance indicates the quality level perceived by the user.Here we measure the round trip delay w.r.t. the packet rate or load.The round-trip delay of the packets was calculated at the source node from the departure time of the packets and the arrival time of their corresponding acknowledgments

The below figure plots the average delay encountered in both CPN and normal IP network.Here the y-axis shows the average delay encountered in milli seconds(ms) and x-axis shows the packet arrival rate into the network(load).

As can be seen in the below graph that average delay encountered by IP network increases as compared to CPN with increase in packet load.This is mainly due to fact that CP are intelligent and adaptive in nature hence are capable of finding of new and suitables routes with the increase in packet load/traffic of a particular roue.But normal IP packets will still follow the same route as they cant change the route themselves.

Also as seen in the below graphs the average delay encountered in normal IP is almost same as CPN till some extent.But as the packet arrival rate increases gradually the delay encountered increases, which can be associated with the fact that in presence of low packet traffic normal IP packets are capable of travelling in the fixed route with high speed same as that of cognitive packets.With increase in packet arrival rate or load the packet traffic increases thus leading to small movement of normal IP packets.CP are able to avoid this by looking for new routes.



<u>Fig-9 Average Delay Vs packet arrival</u>

3.4 Comparision of Package loss in CPN and IP Network

The below figure plots the package loss encountered in both CPN and normal IP network.The packet loss ratio (LR) is calculated from the number of dumb packets received by the destination DPr and the number of dumb packets sent by the source DPs and the number of dumb packets sent by the source DPs.

Here the y-axis shows the package loss (Mbit/s) and x-axis shows the packet arrival into the network .As can be seen in the below graph that package loss in IP significantly high as compared to Cognitive packet Network .

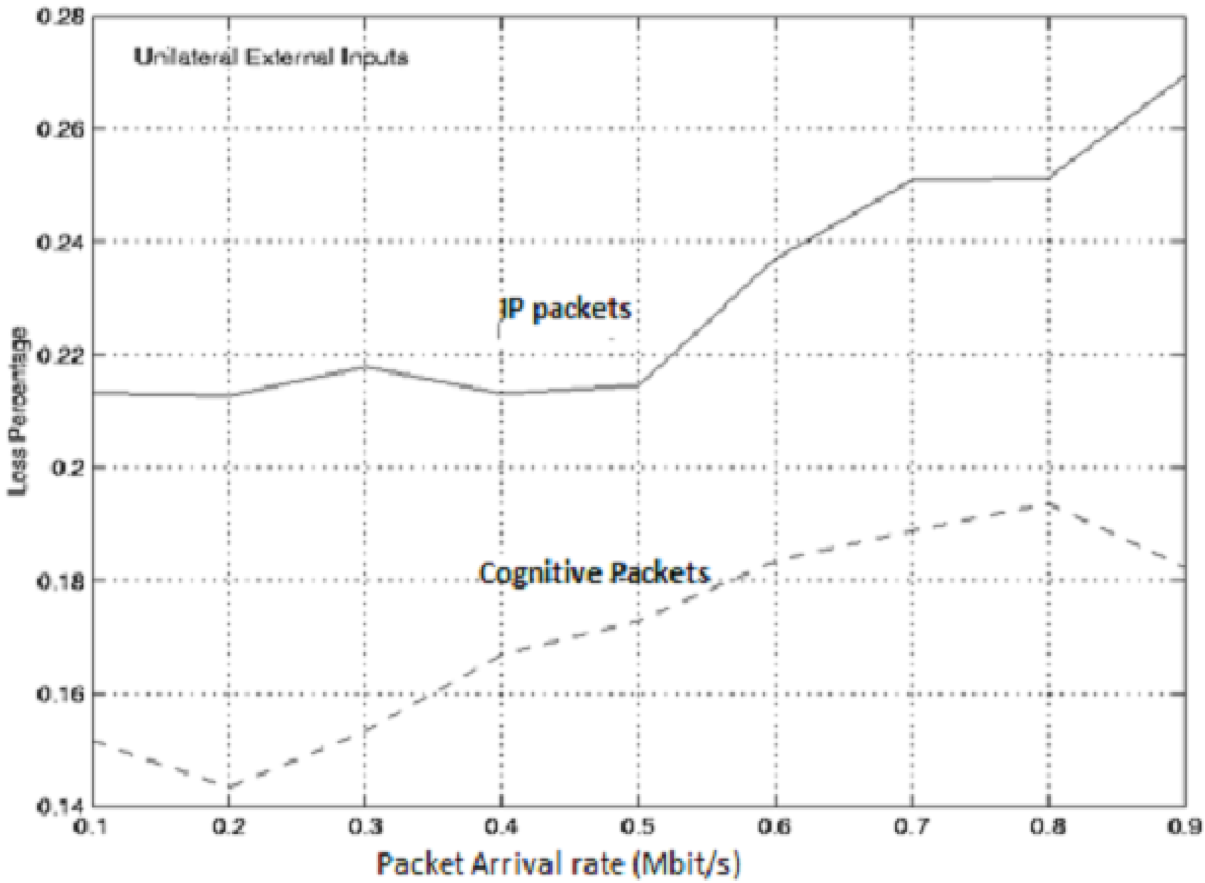


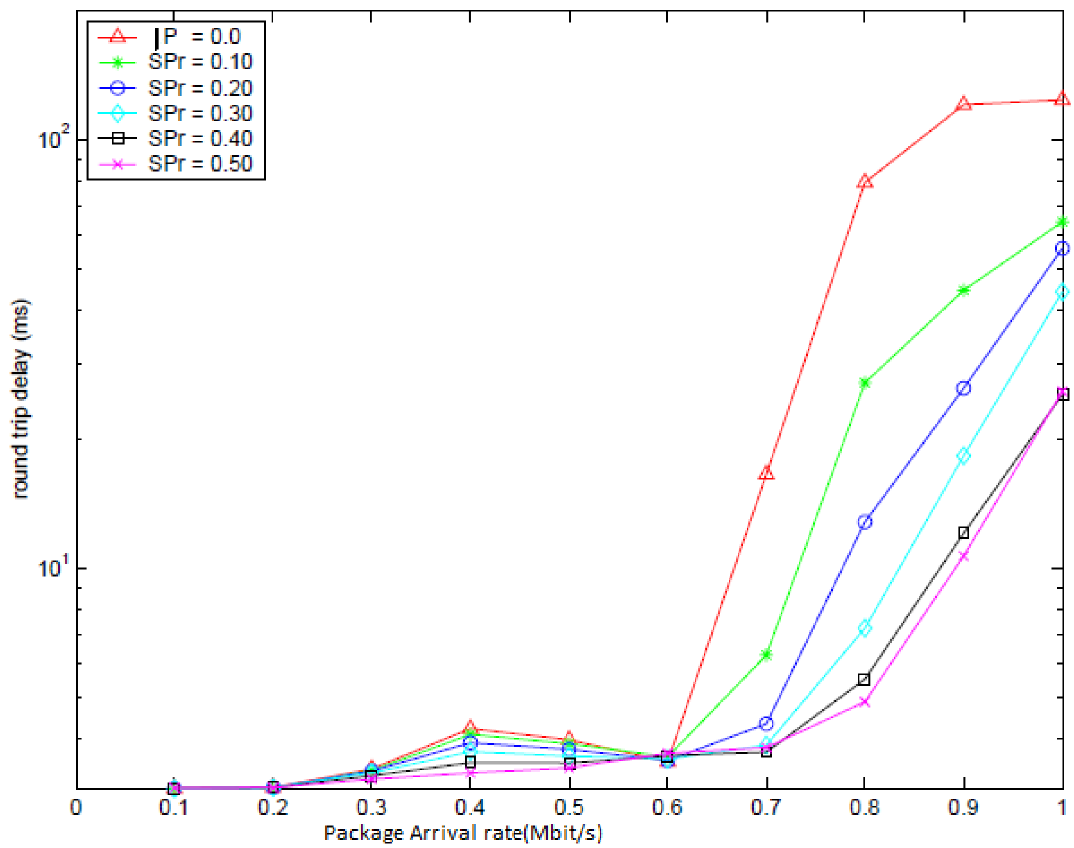
Fig-10 Package loss Vs packet arrival

This is mainly due to the intelligence of cognitive packets. They have knowledge of the network in which they exist and can take routing decisions on their own to avoid any packet loss region. Thus leading to significant decrease in package loss as compared to IP network.

3.5 Simulation result obtained by varying CP percentage

The performance of CPN was found to be changing with change in smart packet ratio of CPN. This has been discussed in the next sections.

Roundtrip delay Vs offered load



<u>**Fig 11 Roundtrip delay Vs offered load**</u>

The round-trip delay of the packets was calculated at the source node from the departure time of the packets and the arrival time of their corresponding acknowledgments.

Fig 11 shows the average round-trip delay of dumb packets as a function of the input traffic ratio (load). As can be seen from the graph with increase in smart packets (higher SP ratio) , the average round-trip delay of dumb packets drops to lower values; this is explained by the fact that the more smart packets, the higher the probability to exploit more paths per unit of time.

Packet loss Ratio Vs offered load

The below graph plots the packet loss and packet load/rate. Here x-axis represents the percentage of lost packets and y-axis shows the package arrival rate/package load.

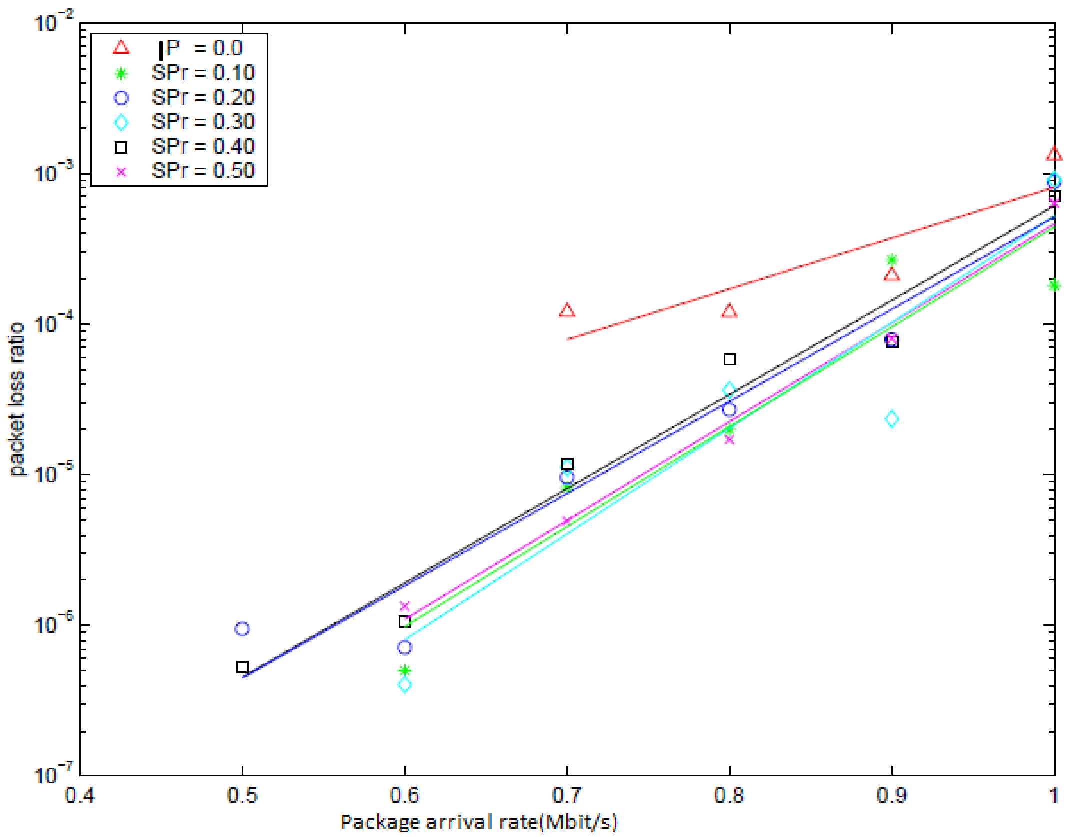


Fig 12 Packet loss Ratio Vs offered load

As can be seen from the graph, packet loss increases with increase in the input load. The packet loss is maximum in normal IP packets it is less a bit less in smart packets. It can also be observed that different smart packet ratios made no appreciable difference in packets loss.

Packet loss during Route Repair

Cognitive packets have learning capability. So they can handle route failure in a much better way as compared to normal IP networks. Due to the learning capability and internal communication between smart packets of a CPN, the link failure or data loss related information is transmitted to all the smart packets once any of them faces it.

Once a smart packet comes to know about link or route failure they look for new routes with their inherent route finding mechanism. As soon as a new route is found this info is communicated to the dumb packets. Which then follow the new route thus leading to the decrease or avoidance of packet loss.

The next figure shows a graph where the packet loss before route recovery is plotted against smart packet ration and logarithm of packet arrival rate.

As can be seen from the graph. The Packet loss is highest in case when the smart packet ratio is zero (normal IP packets). The package loss performance improves with increase the percentage of smart packets in the Cognitive Packet Network which can be associated with the fact that with increase in smart packet ratio the ability to find new routes also increases thus decreasing the packet loss before route recovery.

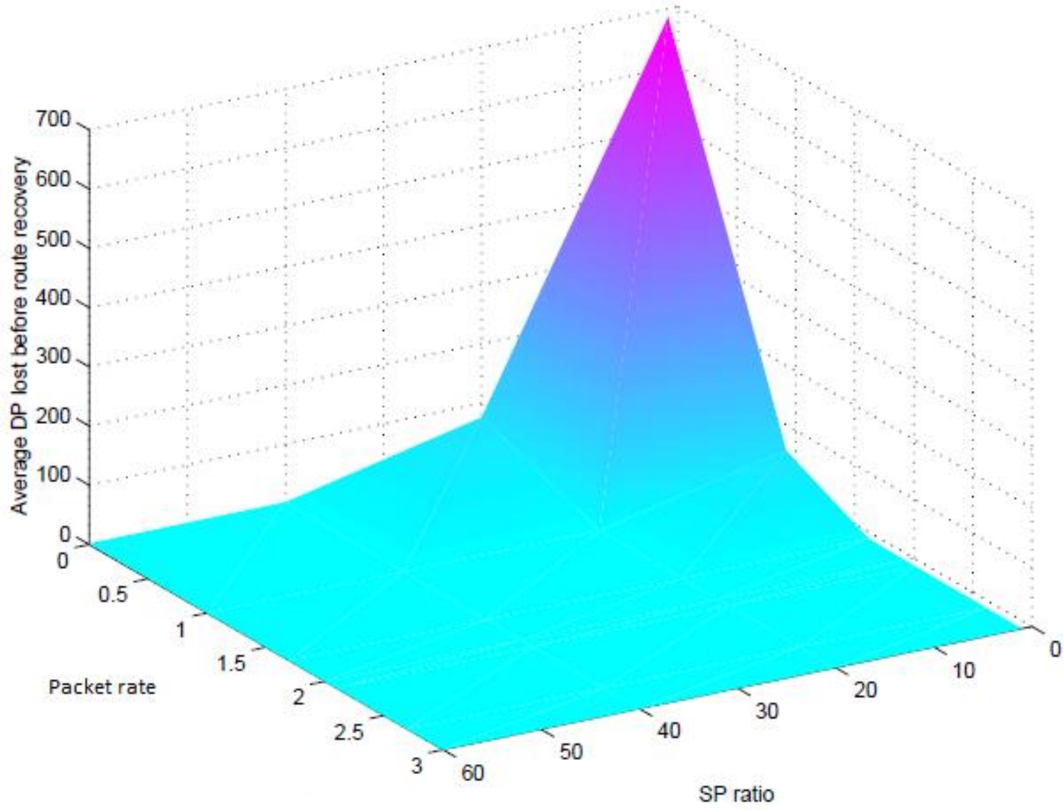


Fig 13 Packet loss during route repair

The packet loss performance improves with increase in packet arrival rate or packet load.

This is mainly due to the fact that at low packet rate/load the number of smart packet will be less. Thus the chances of discovering new routes will be less. With increase in packet rate this improves as the number of smart packet increases.

CHAPTER 4

CONCLUSION AND FUTURE WORK

6.1 Conclusion

Cognitive Packet Networks (CPN) are a new packet network paradigm which address some of the needs of global networking. CPN simplifies router architecture by transferring the control of QOS based best-effort routing to the packet s, away from the routers. Routing tables are replaced by reinforcement-algorithm based routing functions.

CPN is a new packet network paradigm which addresses some of the needs of peer-to-peer networking. CPN transfers the control of QoS based best-effort routing to the connections, which use smart packets for route discovery. Routing tables are replaced by reinforcement algorithm based routing functions. CPN offers “best effort” QoS routing with smart packets both for wire line and wireless networks.

A CPN carries three distinct types of packets: Smart or cognitive packets which search for route based on a QOS driven reinforcement learning algorithm,ACK packets which bring back route information and measurement data from successful smart packets and Dumb packets which do source routing.

In this work we have tried to cover the basic principles of CPNs.Then have tried to describe the Re-enforcement Learning (RL) algorithm which tailors the specific routing algorithm to the QOS needs of a class of packets. And have tried to show the capacity of CPN network to adapt to changes in traffic load and to failures of links.

6.2 Limitations and Future work

Route selection in Cognitive Packet Networks (CPN) occurs continuously for active flows and is driven by the users' choice of a Quality of Service (QoS) goal.
Because routing occurs concurrently to packet forwarding, CPN flows are able to better deal with unexpected variations in network status, while still achieving the desired QoS.

Random neural networks (RNN) play a key role in CPN routing and are responsible to the next-hop decision making of CPN packets. By using reinforcement learning, RNNs' weights are continuously updated based on expected QoS goals and information that is collected by packets as they travel on the network experiencing the current network conditions.

CPN's QoS performance had been extensively investigated for a variety of operating conditions. Its dynamic and self-adaptive properties make them suitable for withstanding availability attacks, such as those caused by worm propagation and denial-of-service attacks.

However, security weaknesses related to confidentiality and integrity attacks have not been examined. This makes them susceptible to integrity and availability attacks.

The inherent self-adaptive properties of the Cognitive Packet Network constitute a significant advantage for helping it withstand attacks against its availability. Its self-adaptation becomes even more effective after introducing packet losses into the goal formulation for training the distributed random neural networks that are used in the routing algorithm.

A denial of service defense mechanism with a given set of detection probabilities becomes more effective if applied in conjunction with the dynamic routing of CPN. Yet, similarly to the Internet Protocol, CPN was designed based on trust. By trusting the information carried by packets and that no node could be compromised, CPN becomes vulnerable to confidentiality and integrity attacks. It offers, by default, little support to ensure end-to-end confidential delivery of data or integrity of packets, which is of crucial importance given that it relies on real-time packets' information to effectively setup and maintain paths.

If we are able to overcome that the benefits from strengthening the confidentiality and integrity, along with the availability of information transmitted through CPN, outweigh the disadvantage of the associated delays that are incurred.

REFRENCES

[1]. Erol Gelenbe, Ricardo Lent, and Zhiguang Xu. Design and performance of cognitive packet networks. Performance Evaluation, 46(2-3): 155-176, 2001.

[2]. Chris Ramming. Cognitive networks. In DARPATech, 2004.

[3]. Nolan, K.E.; Doyle, L.E. "Teamwork and Collaboration in Cognitive Wireless Networks", Wireless Communications, IEEE, On page(s): 22 - 27 Volume: 14, Issue: 4, August 2007.

[4]. Thomas, R.W.; Friend, D.H.; DaSilva, L.A.; MacKenzie, A.B. "Cognitive networks: adaptation and learning to achieve end-to-end performance objectives", Communications Magazine, IEEE,On page(s): 51 - 57 Volume: 44, Issue: 12, Dec. 2006.

[5]. Kärkkäinen, A.P. "Improving situation awareness in cognitive networks using the Self-Organizing Map", Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference on, On page(s): 40 – 47.

[6]. D. D. Clark "A Knowledge Plane for the Internet", Proc. SIGCOMM \'03, pp.3 -10 2003.

[7]. Song, Liang (Jan 2008). "Cognitive Networks: Standardizing the Large Scale Wireless Systems". Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE: 988–992.

[8]. Chris Ramming. Cognitive networks. In DARPATech, 2004.

[9]. P. Gupta, and P. R. Kumar, "Towards an information theory of large networks: an achievable rate region," IEEE Trans. on Information Theory, vol. 49, no. 8, pp. 1877-1894, Aug. 2003.

[10] MARK TURNER 1998. Conceptual Integration Networks. Cognitive Science

22:2 133-187.

[11]. D. Minoli, E. Minoli, “Delivering Voice over IP Network”, John Wiley & Sons, New York, 1998.

[12]. E. Gelenbe (1993) “Learning in the recurrent random neural net-work”, Neural Computation, Vol. 5, No. 1, pp. 154-164, 1993.

[13] E. Gelenbe, Zhi-Hong Mao, Y. Da-Li (1999) “Function approx-imation with spiked random networks” IEEE Trans. on Neural Networks, Vol. 10, No. 1, pp. 3–9, 1999.

[15] E. Gelenbe, E. Seref, Z. Xu “Towards networks with intelligent packets”, Proc. IEEE-ICTAI Conference on Tools for Artificial In-telligence, Chicago, November 9-11, 1999.

[16] U. Halici, “Reinforcement learning with internal expectation for the random neural network” Eur. J. of Opns. Res., Vol. 126, no. 2, pp. 288-307, 2000.

[17] E. Gelenbe, R. Lent, Z. Xu “Towards networks with cognitive packets,” Proc. IEEE MASCOTS Conference, ISBN 0-7695-0728-X, pp. 3-12, San Francisco, CA, Aug. 29-Sep. 1, 2000.

[18] E. Gelenbe, E. S¸eref, Z. Xu, “Simulation with learning agents”, Proceedings of the IEEE, Vol. 89 (2), pp. 148–157, Feb. 2001.

[19] E. Gelenbe and R. Lent “Mobile Ad-Hoc Cognitive Packet Net-works”, Proc. IEEE ASWN, Paris, July 2-4, 2002.

[20] Sakellari, G., June 2009. The Cognitive Packet Network: A Survey. The Computer Journal: Special Issue on Random Neural Networks, doi:10.1093/comjnl/bxp053.

[21] Sakellari, G., 2011. Performance evaluation of the Cognitive Packet Network in the presence of network worms. Performance Evaluation 68 (10), 927–937.

[22] Gelenbe, E., Jul. 2009. Steps toward self-aware networks. Commun. ACM 52 (7), 66–75.URL http://doi.acm.org/10.1145/1538788.1538809

[23] Lent, R., Gelenbe, E., Oct. 2012. Evaluating cpn on a large virtual network testbed. In: Proceedings of the 27th International Symposium on Computer and Information Sciences Series: Lecture Notes in Electrical Engineering. Paris, France.

[24] Sakellari, G., Gelenbe, E., 2010. Demonstrating cognitive packet network resilience to worm attacks.In: ACM Conference on Computer and Communications Security. pp. 636–638.

[25] Oke, G., Loukas, G., 2007. A Denial of Service Detector based on Maximum Likelihood Detection and the Random Neural Network. The Computer Journal 50 (6), 717 – 727.

[26] Oke, G., Loukas, G., September 2008. Distributed Defence Against Denial of Service Attacks: A practical view. In: in Proceedings of the BCS International Conference on Visions of Computer Science. London, UK, pp. 153–162.