Imperial College London

Fast Adaptation with Random Neural Networks

Erol Gelenbe

www.ee.imperial.ac.uk/gelenbe
Denis Gabor Professor
Head of Intelligent Systems and Networks
Dept of Electrical and Electronic Engineering
Imperial College
London SW7 2BT

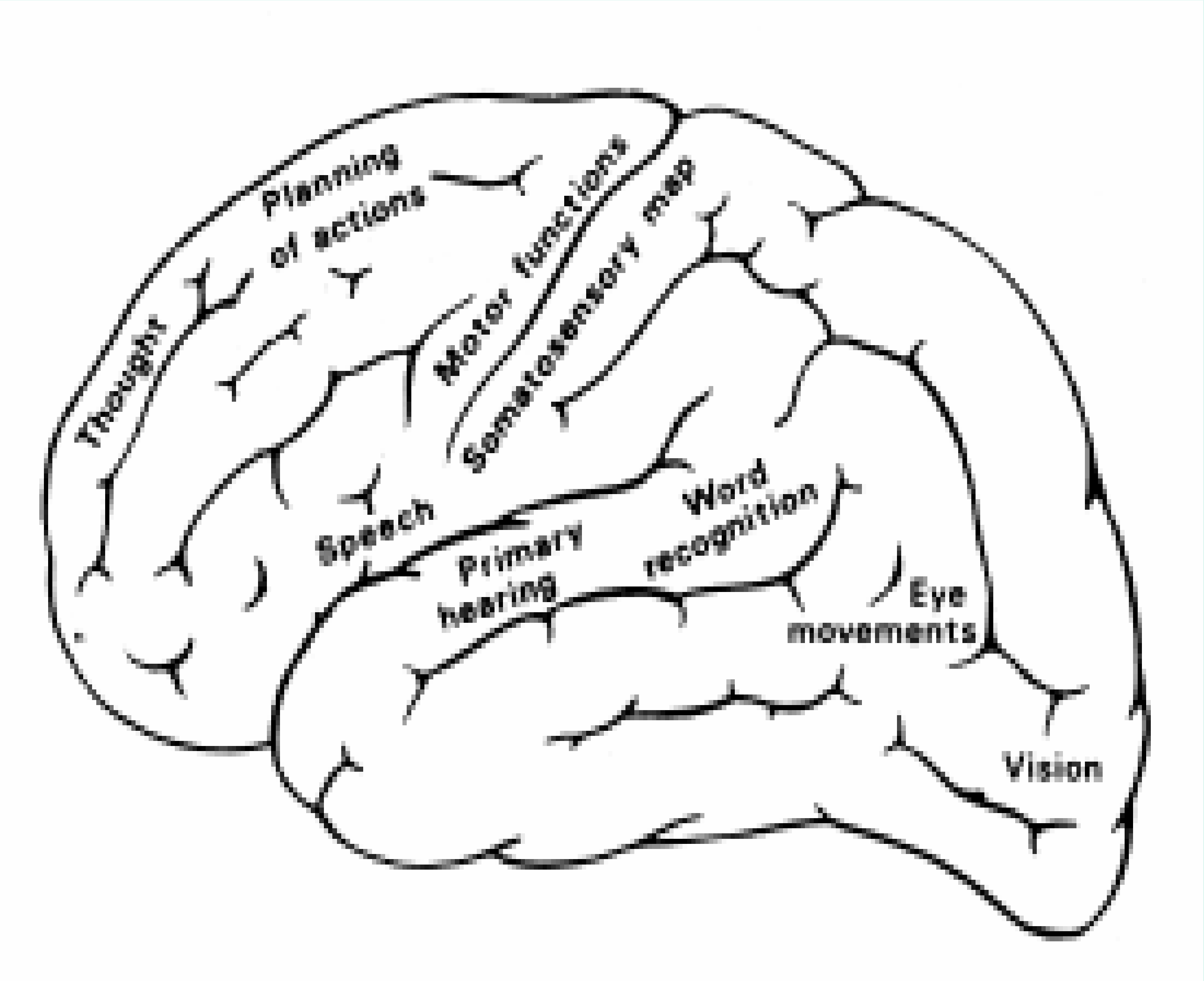
I am grateful to my PhD students who have worked or are working with me on random neural networks and/or their applications

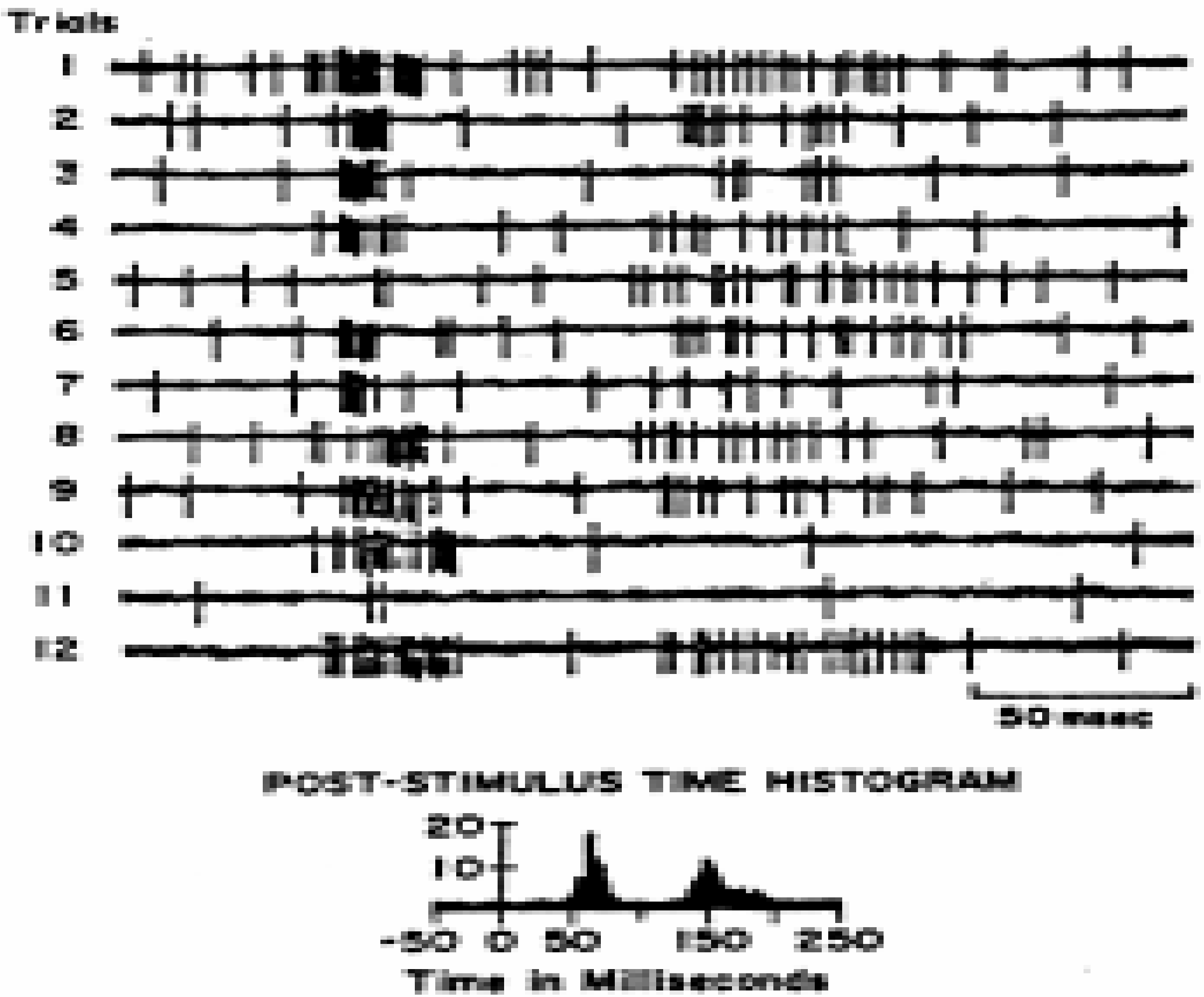
- Univ. of Paris: Andreas Stafylopatis, Jean-Michel Fourneau, Volkan Atalay, Myriam Mokhtari, Vassilada Koubi, Ferhan Pekergin, Jean-Michel Fourneau, Ali Labed, Christine Hubert
- Duke University: Hakan Bakircioglu, Anoop Ghanwani, Yutao Feng, Chris Cramer, Yonghuan Cao, Hossam Abdelbaki, Taskin Kocak
- UCF: Rong Wang, Pu Su, Peixiang Liu, Will Washington, Esin Seref, Zhiguang Xu, Khaled Hussain, Ricardo Lent
- Imperial: Arturo Nunez, Varol Kaptan, Mike Gellman, Georgios Loukas, Yu Wang

Thank you to the agencies and companies who have supported my RNN work generously over the last 15 yrs

- France (1989-97): ONERA, CNRS C3, Esprit Projects QMIPS, EPOCH and LYDIA
- USA (1993-): ONR, ARO, IBM, Sandoz, US Army Stricom, NAWCTSD, NSF, Schwartz Electro-Optics, Lucent
- UK (2003-): EPSRC, MoD, General Dynamics UK Ltd, EU FP6 for grant awards for the next three years, hopefully more ..

Random Spiking Behaviour of Neurons





Trials
1
2
3
4
5
6
7
8
9
10
11
12
50 msec
POST-STIMULUS TIME HISTOGRAM
20
10
-50 0 50 150 250
Time in Milliseconds

Random Spiking Behaviour of Neurons

Work started as an individual basic research project, motivated by a critical look at modeling biological neurons, rather than using popular connectionist models

Biological characteristics of the model needed to include:

-Action potential "Signals" in the form of spikes of fixed amplitude

-Modeling recurrent networks

-Random delays between spikes

-Conveying information along axons via variable spike rates

-Modeling different signals, e.g. electrical <u>and</u> chemical

-Store and fire behaviour of the soma (head of the neuron)

-Reduction of neuronal potential after firing

-Possibility of representing axonal delays between neurons

-Arbitrary network connectivity

Random Spiking Behaviour of Neurons

Mathematical properties that we hoped for, but did not always expect to obtain, but which were obtained

-Existence and uniqueness of solution to the models
-Closed form analytical solutions for large systems
-Convergent learning for recurrent networks
-Polynomial speed for recurrent gradient descent
-Hebbian and reinforcement learning algorithms
-Analytical annealing

We exploited the analogy with queuing networks, and this also opened a new chapter in queuing network theory now called "G-networks" where richer models were developed

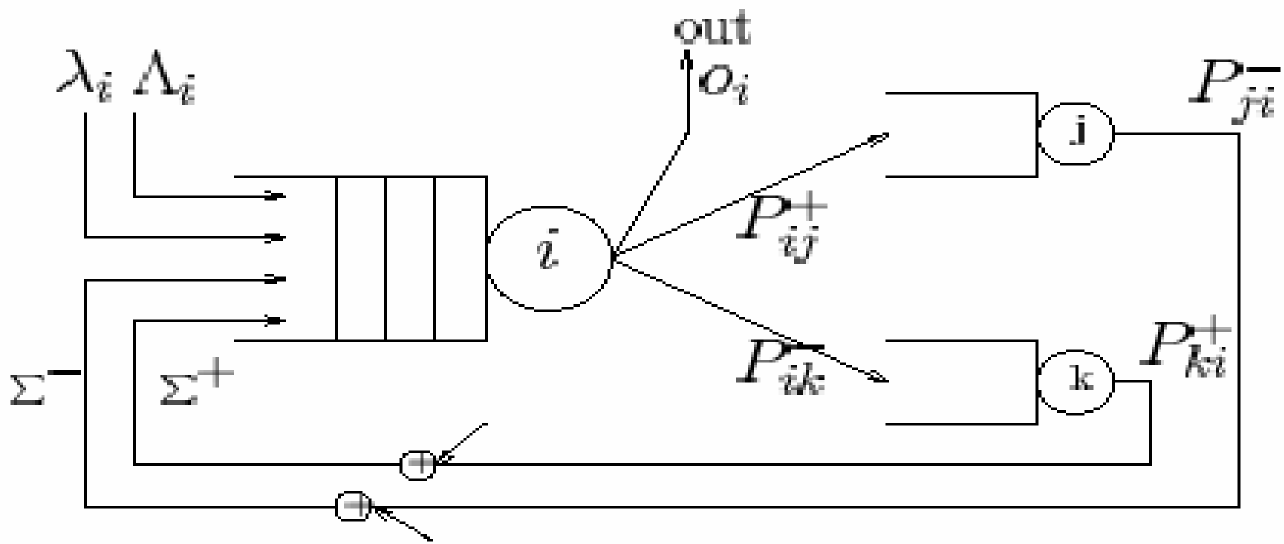
Queuing Networks: Exploiting the Analogy

-Discrete state space, typically continuous time, stochastic models arising in studying populations, dams, production systems, communication networks ..
-Important theoretical foundation for computer systems performance analysis
-Open (external Arrivals and Departures), as in Telephony, or Closed (Finite Population) as in Compartment Models
-Systems comprised of Customers and Servers
-Theory is over 100 years old and still very active .. e.g. the Swedish Academy of Science's Mathematics Division has a full year invited program in 2004 on the subject (sniff sniff, I am going to Stockholm from 11/15/04 for one month !)
-Big activity at Bell Labs, AT&T Labs, IBM Research
- More than 100,000 papers on the subject ..

Queuing Network
Random Neural Network

-Open (external Arrivals and Departures), as in Telephony, or Closed (Finite Population) as in Compartment Models
-Systems comprised of Customers and Servers
-Servers = Neurons
-Customer .. Arriving to server will increase the queue length by +1
-Excitatory spike arriving to neuron will increase its soma's potential by +1
-Service completion (neuron firing) at server (neuron) will send out a customer (spike), and reduce queue length by 1
-Inhibitory spike arriving to neuron will decrease its soma's potential by -1
-Spikes (customers) leaving neuron i (server i) will move to neuron j (server j) in a probabilistic manner

The RNN



- This is a spiked neural network model .. excitation spikes "+1" and inhibition spikes "-1" travel in the newtork
- The state of neuron i is a non-negative integer k_i
- The state of the n-neuron network is a vector $(k_1, ..., k_n)$

Mathematical Model: A "neural" network with n neurons

Internal State of Neuron i, is an Integer $x_i \geq 0$

Network State at time t is a Vector

$x(t) = (x_1(t), \dots , x_i(t), \dots , x_k(t), \dots , x_n(t))$

Is the Internal Potential of Neuron I

If $x_i(t) > 0$, we say that Neuron i is excited and it may fire at t^+ in which case it will send out a spike

If $x_i(t)=0$, the Neuron cannot fire at t^+

When Neuron i fires: :

- It sends a spike to some Neuron k, w.p. p_{ik}
- Its internal state changes $x_i(t^+) = x_i(t) - 1$

State of Network

$x(t) = (x_1(t), \ldots, x_i(t), \ldots, x_l(t), \ldots, x_n(t)),\ x_i(t) \geq 0$

If $x_i > 0$, we say that Neuron i is excited

f $x_i(t) > 0$, then Neuron i will fire with probability $r_i \Delta t$ in the interval $[t, t+\Delta t]$, and as a result:

Its internal state changes $x_i(t^+) = x_i(t) - 1$

It sends a spike to some Neuron m w.p. p_{im}

The arriving spike at Neuron m is an

- Excitatory Spike w.p. p_{im}^+
- Inhibitory Spike w.p. p_{im}^-
- $p_{im} = p_{im}^+ + p_{im}^-$ with $\Sigma^n_{m=1} p_{im} \leq 1$ for all i=1,..,n

Rates and Weights

$x(t) = (x_1(t), \ldots, x_i(t), \ldots, x_l(t), \ldots, x_n(t)), x_i(t)>0$

f $x_i(t) > 0$, then Neuron i will fire with probability

$r_i \Delta t$ in the interval $[t, t+\Delta t]$, and as a result:

From Neuron i to Neuron l

- Excitatory Weight or Rate is $w_{im}^+ = r_i\, p_{im}^+$
- Inhibitory Weight or Rate is $w_{im}^- = r_i\, p_{im}^-$
- Total Firing Rate is $r_i = \Sigma^n_{m=1} w_{im}^+ + w_{im}^-$

To Neuron i, from Outside the Network

- External Excitatory Spikes arrive at rate Λ_i
- External Inhibitory Spikes arrive at rate λ_i

State Equations

$p(k,t) = \Pr[x(t) = k]$ where $\{x(t): t \geq 0\}$ is a discrete state-space Markov process,

and $k_{ij}^{+-} = k + e_i - e_j$, $k_{ij}^{++} = k + e_i + e_j$

$k_i^{+} = k + e_i$, $k_i^{-} = k - e_i$:

The **Chapman** - Kolmogorov Equations

$$\frac{d}{dt} p(k,t) = \sum_{i,j} [p(k_{ij}^{+-},t) r_i p_{ij}^{+} 1[k_j(t) > 0] + p(k_{ij}^{++},t) r_i p_{ij}^{-}] + \sum_i [p(k_i^{+},t)(l_i + r_i d_i) + \Lambda_i p(k_i^{-},t) 1[k_i(t) > 0]]$$

$$- p(k,t) \sum_i [(l_i + r_i) 1[k_i(t) > 0] + \Lambda_i]$$

Let :

$$p(k) = \lim_{t \to \infty} \Pr[x(t) = k], \quad \textit{and} \quad q_i = \lim_{t \to \infty} \Pr[x_i(t) > 0]$$

Theorem *If the* $C - K$ *equations have a stationary solution,*

then it has the "product – form" $p(k) = \prod_{i=1}^{n} q_i^{k_i} (1 - q_i)$, *where*

External Arrival
Rate of Excitatory
Spikes

Probability that
Neuron I is excited

$$0 \le q_i = \frac{\Lambda_i + \sum_j q_j \overbrace{r_j p^{+}_{ji}}^{w^{-}_{ji}}}{r_i + l_i + \sum_j q_j \underbrace{r_j p^{-}_{ji}}_{w^{+}_{ji}}} < 1$$

Firing Rate of
Neuron i

External Arrival
Rate of Inhibitory
Spikes

Theorem (Gelenbe 93, Gelenbe - Schassberger 95)

The system of non – linear equations

$$q_i = \frac{\Lambda_i + \sum_j q_j r_j p_{ji}^{+}}{r_i + l_i + \sum_j q_j r_j p_{ji}^{-}} \quad , \quad 1 \le i \le n$$

has an unique solution if all the $q_i < 1$.

Theorem (Gelenbe et al. 99) *Let* $g : [0,1]^v \to R$ *be continuous and bounded. For any* $e > 0$, *there exists an RNN with two output neurons* q_{o+}, q_{o-} *s.t.*

$$\sup_{x \in [0,1]^v} | g(x) - y(x) | < e \quad \text{for} \quad y(x) = \frac{q_{o+}}{1 - q_{o+}} - \frac{q_{o-}}{1 - q_{o-}}$$

Cortico-Thalamic Response to Somato-Sensory Input
Or ….
What Does the Rat Think when you Tweak Her/His Whisker?

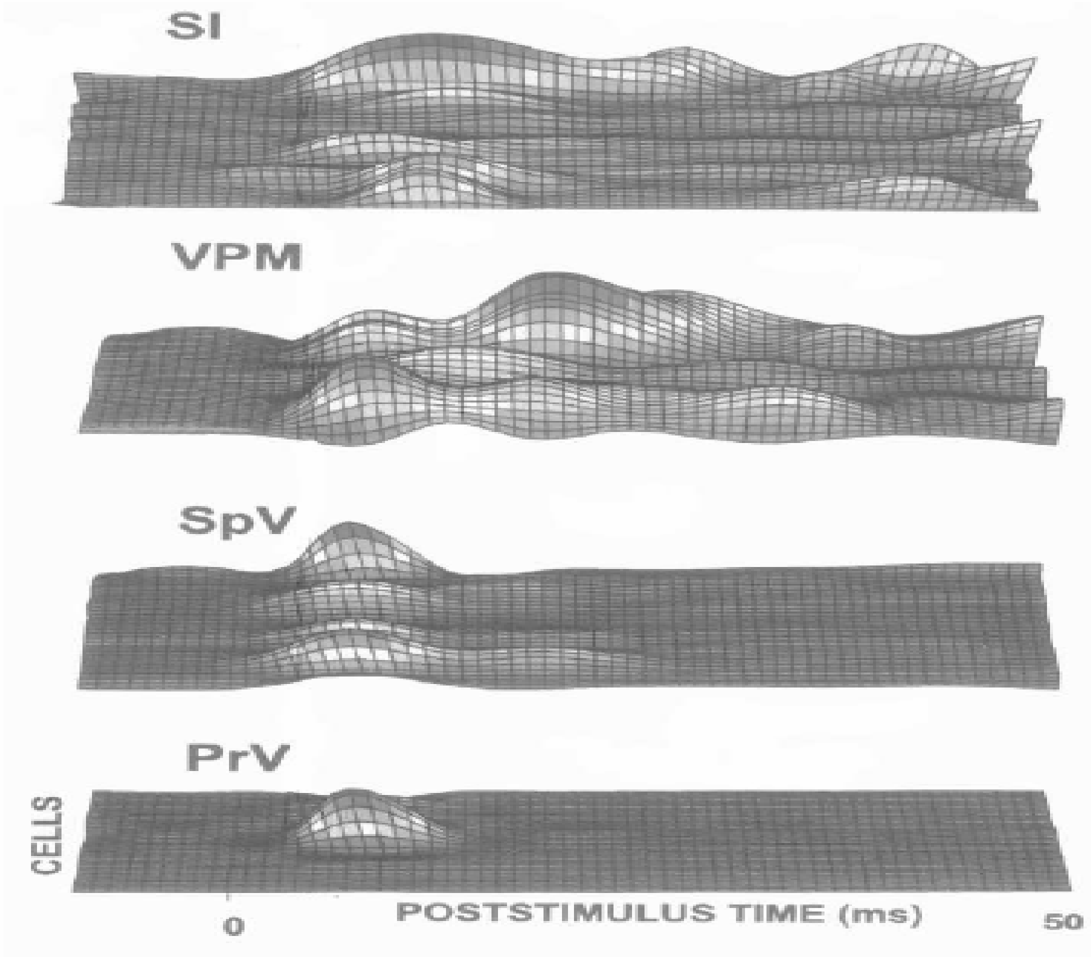


Figure 1: Input from the brain stem (PrV) and response at thalamus (VPM) and cortex (SI), reprinted from M.A.L. Nicolelis et al. "Recontructing the engram: simultaneous, multiple site, many single neuron recordings", *Neuron* vol. 18, 529-537, 1997.

Rat Brain Modeling with the Random Neural Nework

- Clarify Some of the Mechanisms which Influence (Brain) Cortico-Thalamic Oscillations

- Start with Oscillations Observed in a Physiologically well understood system: the Rat "Barrel Neurons"

- Use a Recurrent "Random Network (RNN)" Spiked Model which actually Models the (Observed) Natural Neurons' Spiked Behaviour

- Identify Primary Factors Causing Oscillations

- We propose a theoretical model that can provide insight into the nature of the observed response in SpV, VPM and SI which (cf. Nicolelis) “cannot be defined as discrete representations of the cutaneous periphery”.

- The purpose of the model is to investigate how the individual neuron characteristics, and the network architecture that connects the layers, and the cells within each layer, impacts the observed response.

- The model is composed of three schematic layers which represent the physiologically identified layers in the rat: T (thalamus), R (reticular layer) and C (cortex) with the excitatory (+) and inhibitory (-) connections and feedback loops shown in Figure 3.

- Feedback loops – both excitatory and inhibitory – are present both inside cortex and between the various ensembles of cells.

The Biological Model

- The somatosensory stimulus (involving a single whisker of the rat) impacts a physiological system in which the number of thalamic cells T is of the order of 10^3, while 10^2 cortical cells C are involved.

- The model assumes that all cortical cells involved are statistically identical, that all thalamic cells are statistically identical, and that all reticular layer cells are also statistically identical.

- In relation to Simons et al., thalamic cells T correspond to thalamo-cortical units (TCU), cortical cells C correspond to "regular spike" barrel units (RSU) ofsomatosensory cortex.

But Back to the Rat’s Whiskers and Nicollelis’ and Others’ Measurements

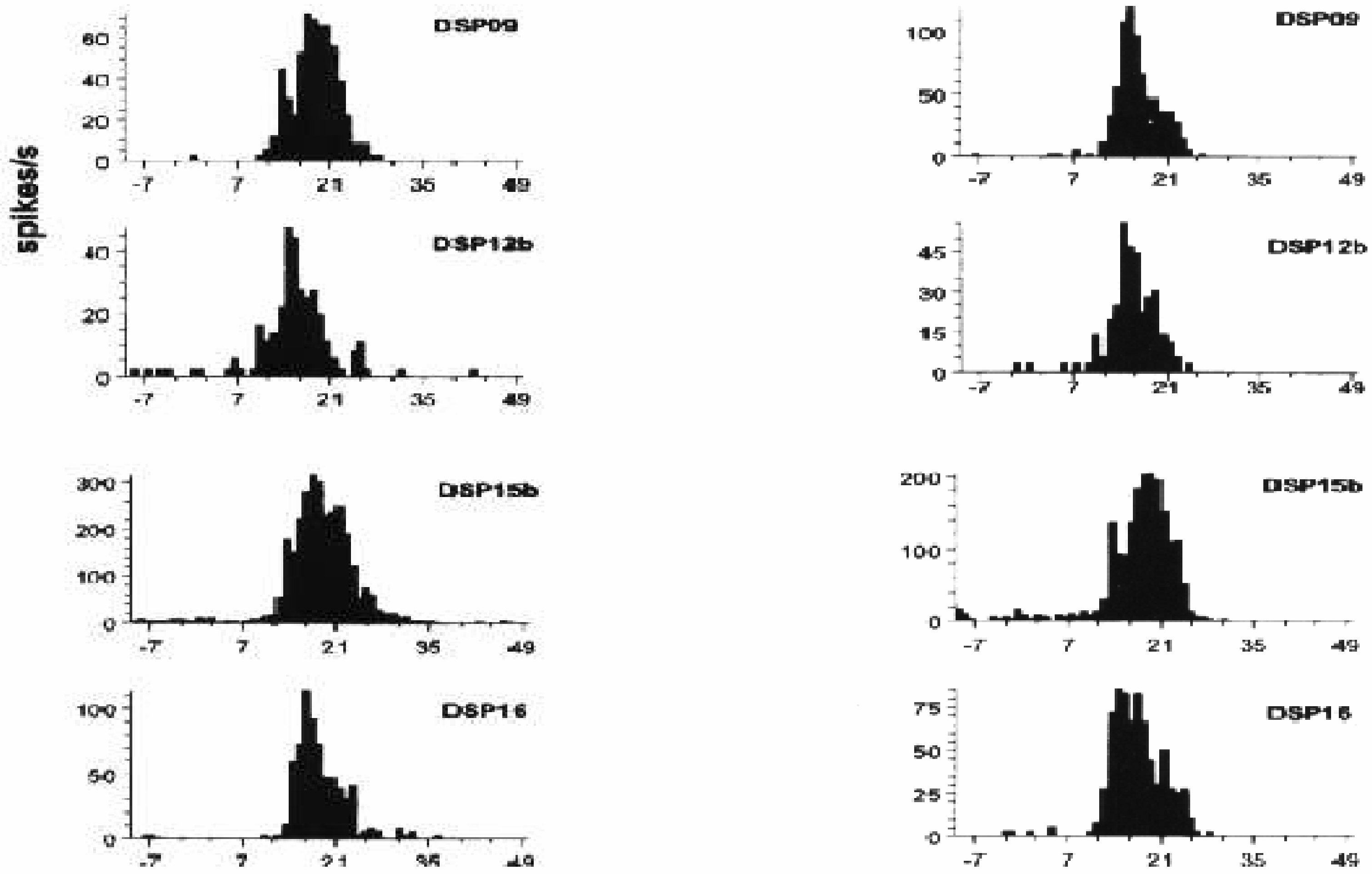
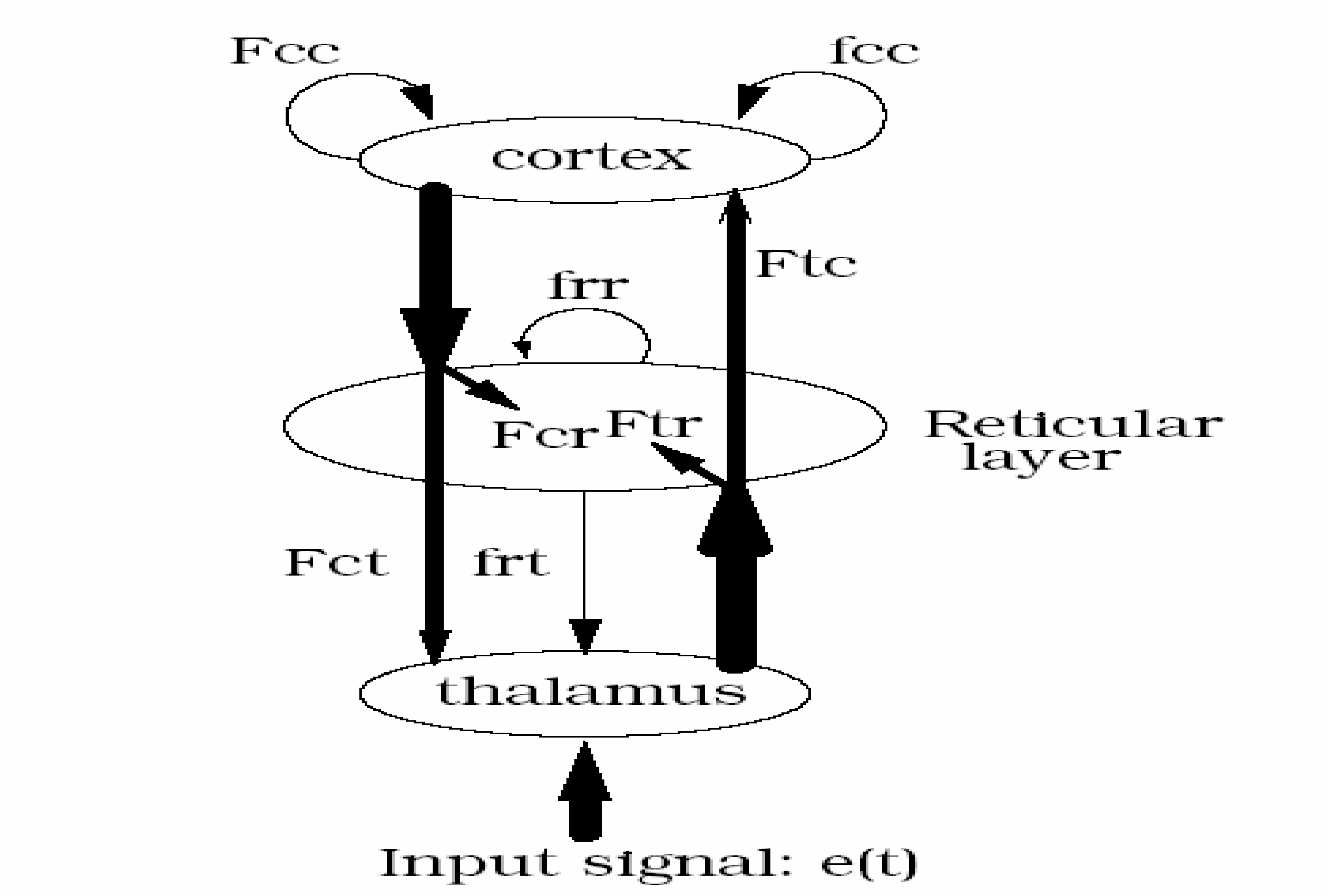


Figure 2: Distinct cell poststimulus firing, reprinted from M.A.L. Nicolelis et al. “Reconstructing the engram: simultaneous, multiple site, many single neuron recordings”, *Neuron* vol. 18, 529-537, 1997.

Network Architecture from Physiological Data

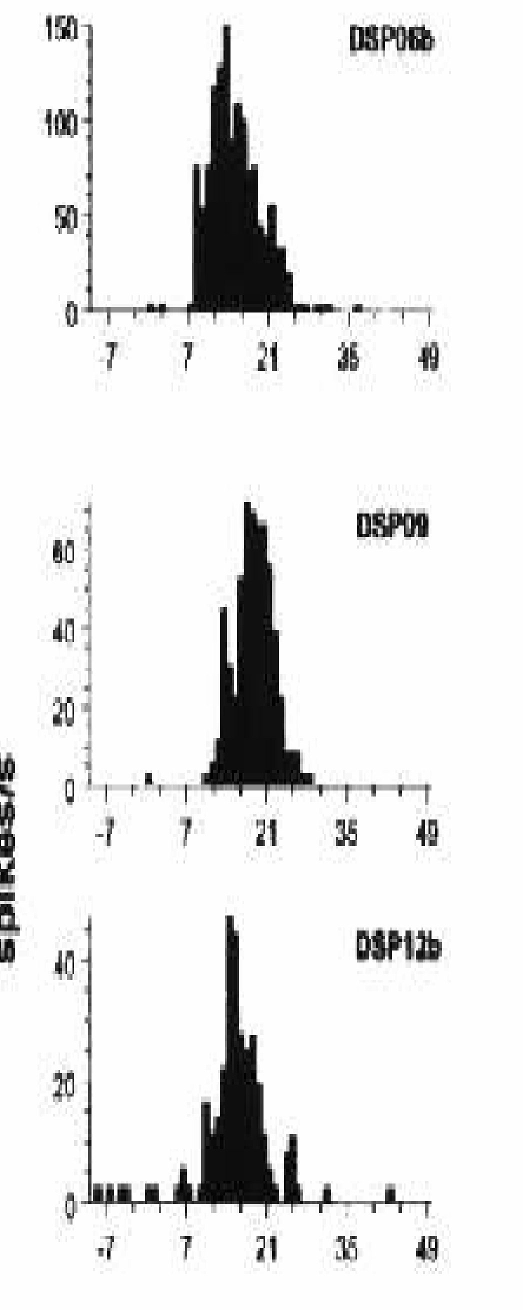


- The maximum cortex cell firing rate observed in the measure- ments in Figure 2 ranges from 40 to 300 pulses per second o 0.04 to 0.3 pulses/ms. We take an intermediate value in thi range for our numerical examples, setting $r_c = 0.1$. The choic of $D_{cc} = D_{ct} = D_{cr} = 10$ is compatible with the value se lected for r_c, since for the RNN model the choice of r_c implie that (a) when a cortex neuron is excited it fires on the averag each r_c^{-1} milliseconds, and (b) a cortex neuron has an averag latency of r_c^{-1} milliseconds between the arrival at its input o signals of sufficient strength to excite it, and the moment i starts emitting its first output spike.

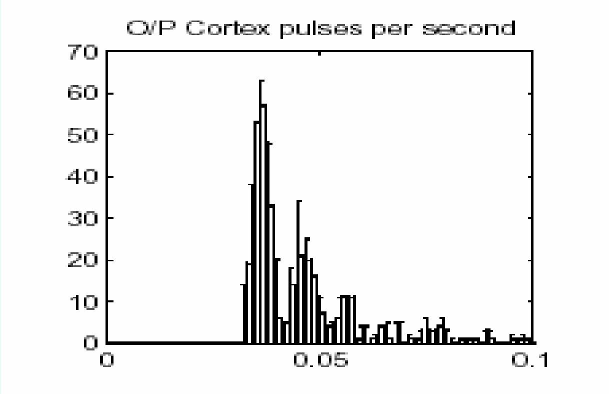
- Assuming that maximum firing rates in thalamus (VPM) ar some ten times higher than in cortex, as are firing rates in th reticular area, we take $r_t,\ r_r = 1$.

- We have assumed that 20% of cortex to cortex connection are inhibitory, while 80% are excitatory; this is consistent wit known data (Steriade '90) who reports at most 25% cortex t cortex inhibitory connections in certain species of monkeys).

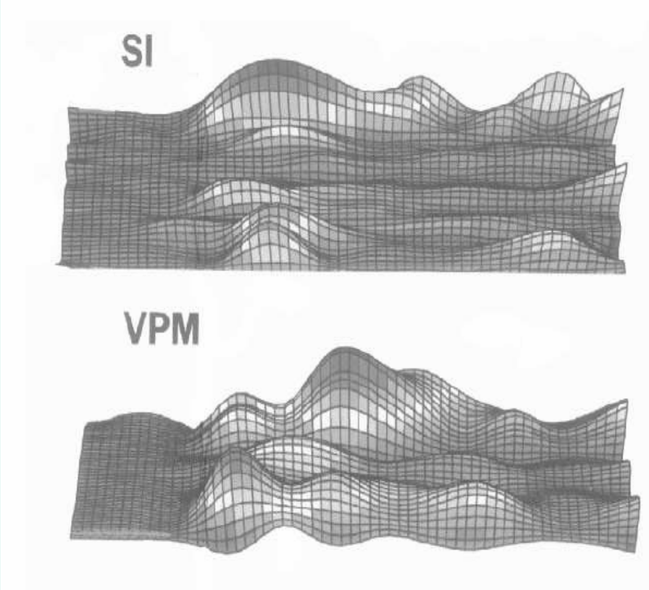
Comparing Measurements and Theory: Calibrated RNN Model and Cortico-Thalamic Oscillations



Single Cell Recordings (Nicollelis et al '97)



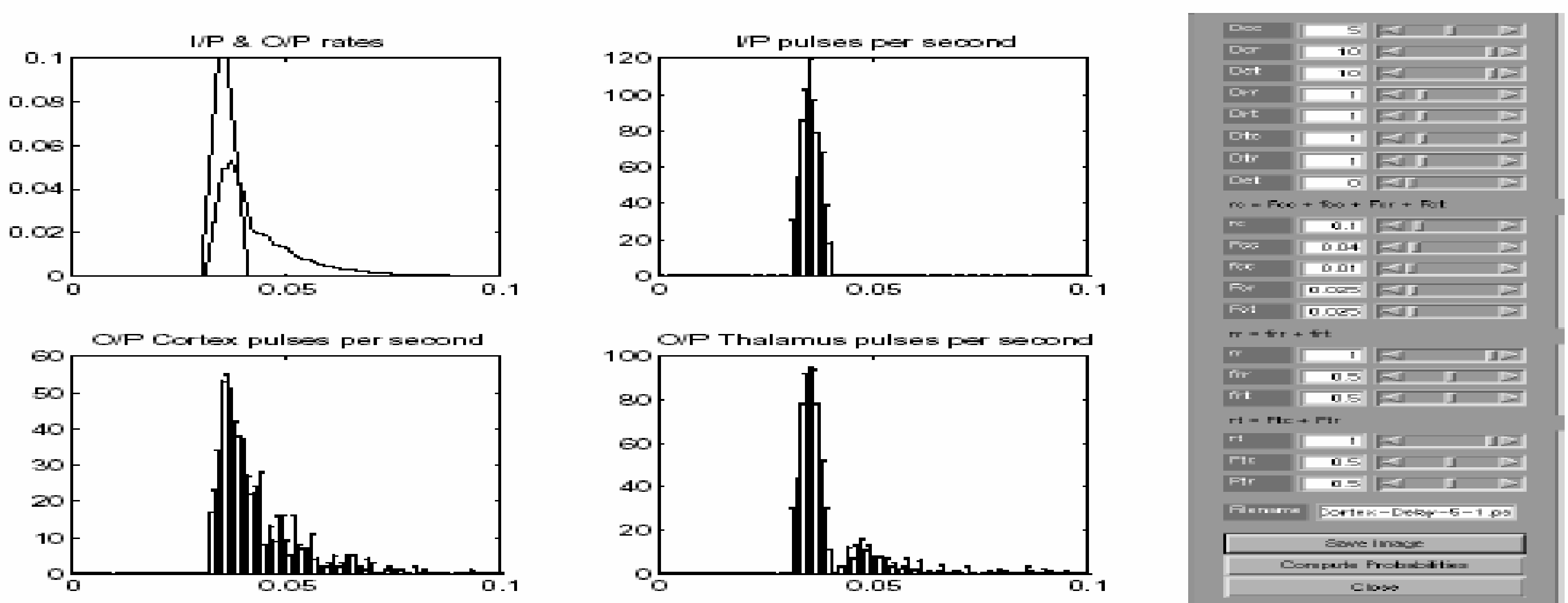
Predictions of Calibrated RNN Mathematical Model (Gelenbe & Cramer '98, '99)



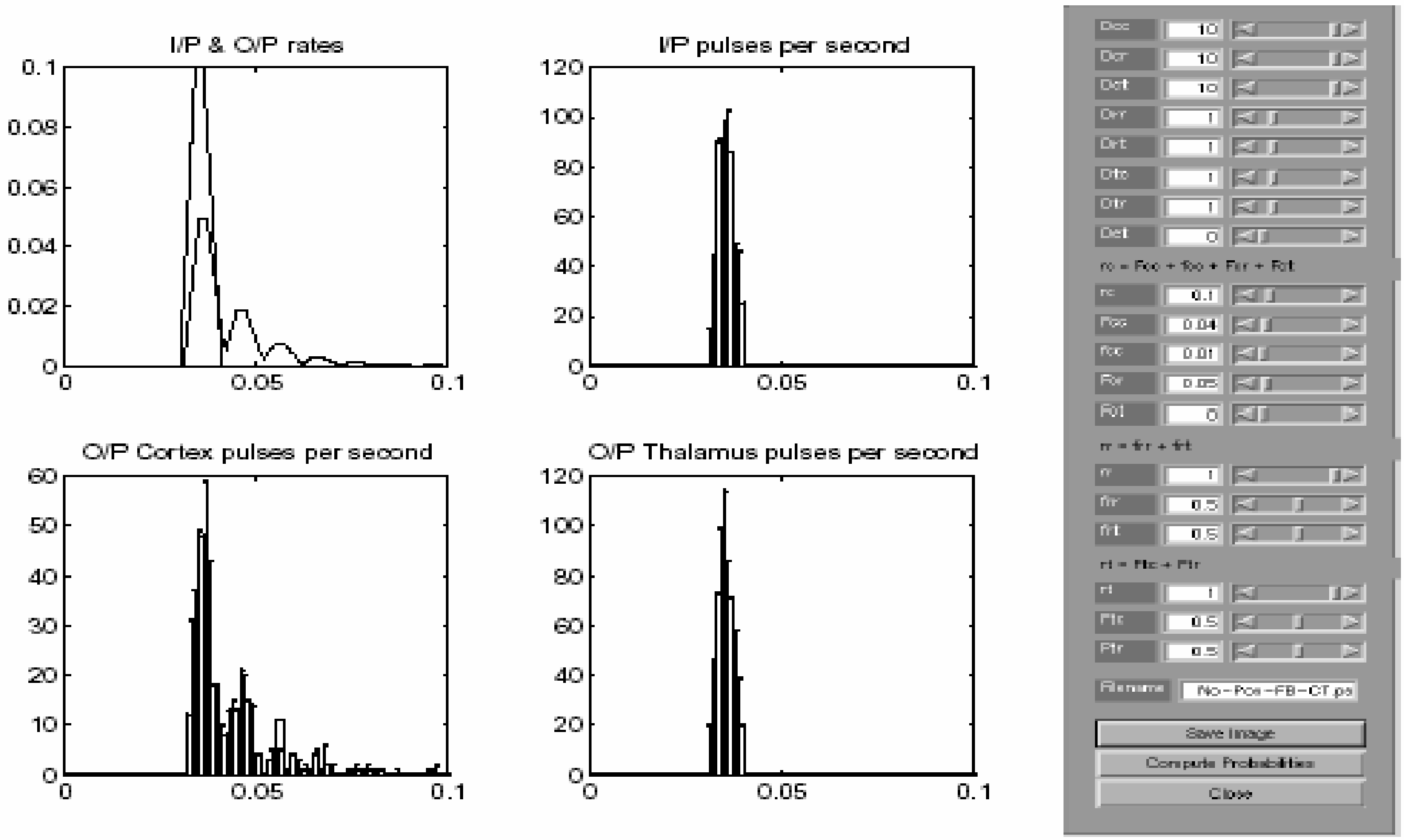
Simultaneous Multiple Cell Recordings (Nicollelis et al.)

Response parameters and oscillatory behavior are consistent in the model for cortex with observations, and the duration of oscillations (circa 50 ms) are comparable to but perhaps slightly greater than those observed. Another corroboration of model predictions concerns the observed latencies to peak response in cortical cells which are reported by Simons et al. to be of just under $12ms$. This is very consistent with the model predictions where peak response is observed some $10\ to\ 15ms$ after onset of the stimulus.

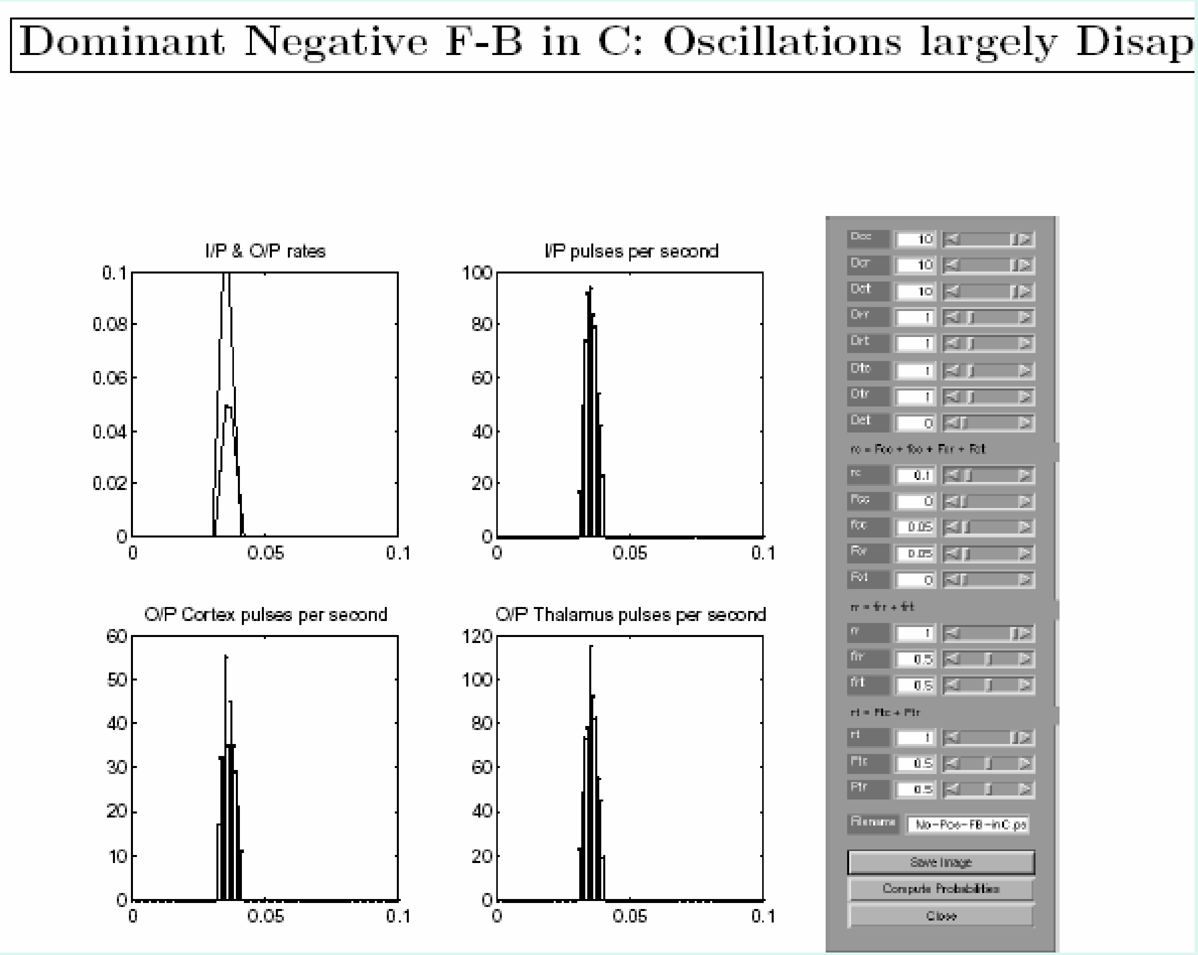
5/1 Cortex/Thalamus Delay: Oscillations Disappear



No Pos. F-B from C to T: C Oscillates, T does Not



When Feedback in Cortex is Dominantly Negative, Cortico-Thalamic Oscillations Disappear



- Positive Feedback loops within cortex significantly affect the existence of the damped oscillatory phenomenon, and its duration.

- Positive cortex to thalamus Feedback is not needed for cortical oscillations, but is needed for thalamic oscillations.

- Cortex to thalamus negative Feedback via the reticular layer, and cortex to cortex inhibitory connections, contribute to damping.

- The reticular layer affects the amplitude of the oscillations, but not their causes. Projections from thalamus to cortex reduce the amplitude but do not modify the damping constants or periods of the oscillations.

Developments

- Learning techniques: Hebbian, Gradient Based, Reinforcement Learning
- Analytical (Simulated) Annealing
- Applications to Image Processing
- New Mathematical Developments and Extensions
- Applications to QoS in the Internet:

  - Video Compression and Video Traffic Shaping,

  - QoS Driven Routing Protocols and CPN

Gradient Computation for the Recurrent RNN is O(n³)

Let $\mathbf{q} = (q_1, ..., q_n)$, and define the $n \times n$ matrix

$$\mathbf{W} \;=\; \{[w^+(i,j) - w^-(i,j)q_j]/\lambda^-(j)\} \quad i,j = 1, ..., n$$

The vector equations can now be written as:

$$\begin{aligned}
\partial\mathbf{q}/\partial w^+(u,v) &= \partial\mathbf{q}/\partial w^+(u,v)\mathbf{W} + \gamma^+(u,v)q_u \\
\partial\mathbf{q}/\partial w^-(u,v) &= \partial\mathbf{q}/\partial w^-(u,v)\mathbf{W} + \gamma^-(u,v)q_u
\end{aligned}$$

where the elements of the n-vectors $\gamma^+(u,v) = [\gamma_1^+(u,v), \ldots, \gamma_n^+(u,v)]$ and $\gamma^-(u,v) = [\gamma_1^-(u,v), \ldots, \gamma_n^-(u,v)]$ are

$$\gamma_i^+(u,v) = \begin{cases} -1/\lambda^-(i) & \text{if } u = i, v \neq i \\ +1/\lambda^-(i) & \text{if } u \neq i, v = i \\ 0 & \text{for all other values of } (u,v) \end{cases}$$

$$\gamma_i^-(u,v) = \begin{cases} -(1+q_i)/\lambda^-(i) & \text{if } u = i, v = i \\ -1/\lambda^-(i) & \text{if } u = i, v \neq i \\ -q_i/\lambda^-(i) & \text{if } u \neq i, v = i \\ 0 & \text{for all other values of } (u,v) \end{cases}$$

Notice that

$$\begin{aligned}
\partial\mathbf{q}/\partial w^+(u,v) &= \gamma^+(u,v)q_u[\mathbf{I} - \mathbf{W}]^{-1} \\
\partial\mathbf{q}/\partial w^-(u,v) &= \gamma^-(u,v)q_u[\mathbf{I} - \mathbf{W}]^{-1}
\end{aligned} \qquad (4)$$

where $\mathbf{I}$ denotes the n by n identity matrix. Hence the main computational effort in this algorithm is to obtain $[\mathbf{I} - \mathbf{W}]^{-1}$.

Random Neural Network

Neurons exchange Excitatory and Inhibitory Spikes (Signals)

Inter-neuronal Weights are Replaced by Firing Rates

Neuron Excitation Probabilities obtained from Non-Linear State Equations

Steady-State Probability is Product of Marginal Probabilities

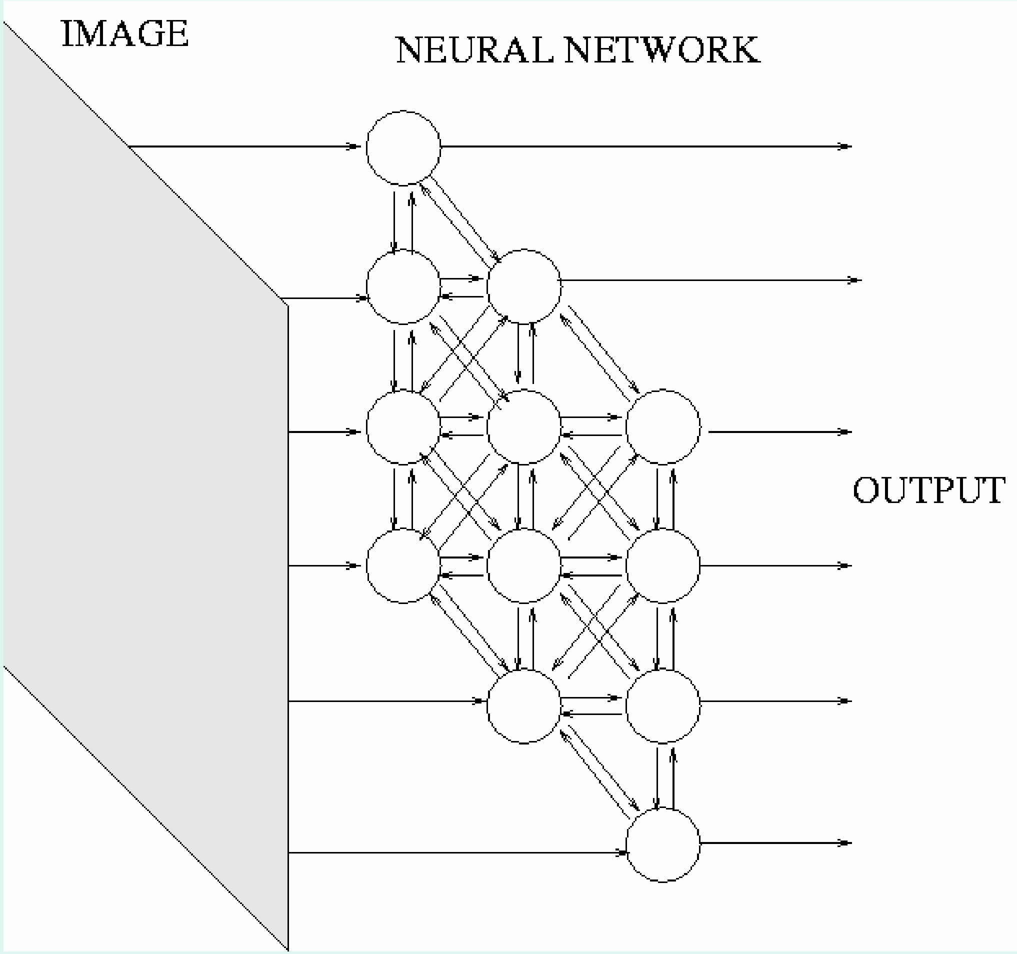
Separability of the Stationary Solution based on Neuron Excitation Probabilities

Existence and Uniqueness of Solutions for Recurrent Network

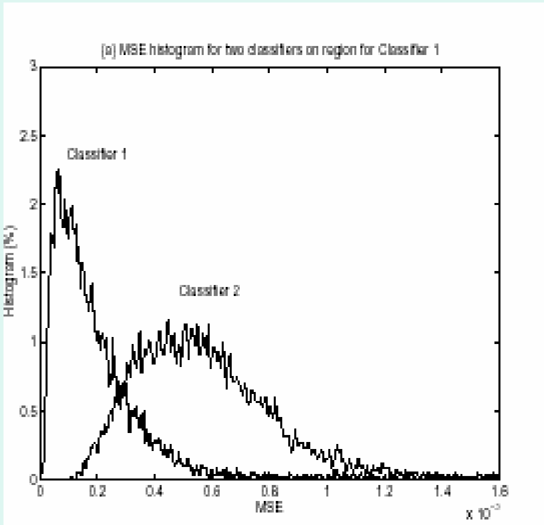
Learning Algorithms for Recurrent Network are $O(n^3)$

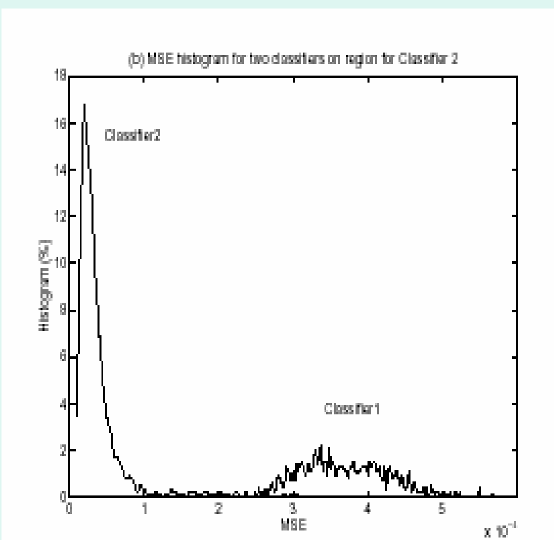
Multiple Classes (1998) and Multiple Class Learning (2002)

Texture Based Object Identification Using the RNN, US Patent ’99 (E. Gelenbe, Y. Feng)



1) MRI Image Segmentation





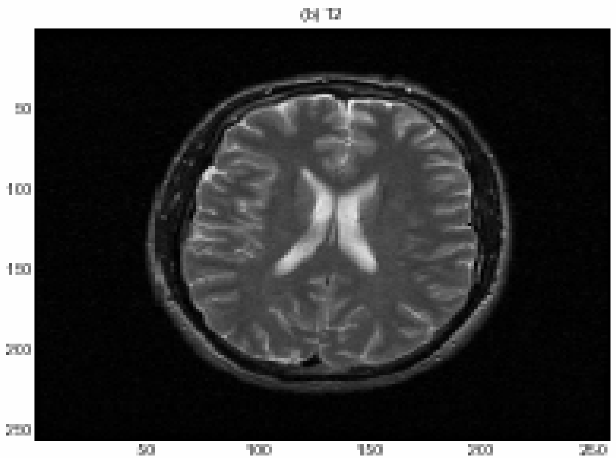


Figure 7: A T2 MR image before being processed by our method

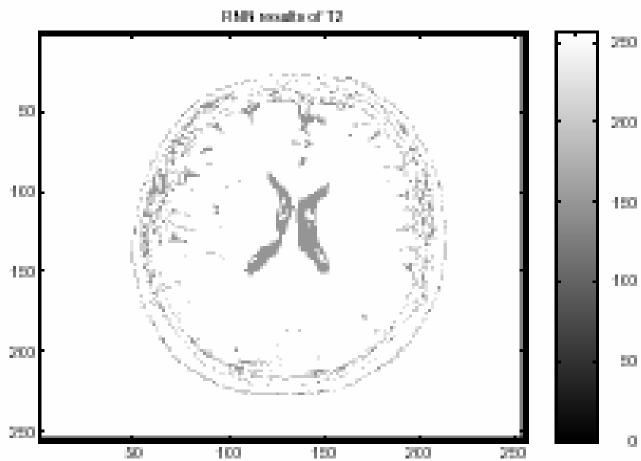


Figure 8: Result of RNN processing for the T2 MRI image.

Brain Image Segmentation with RNN

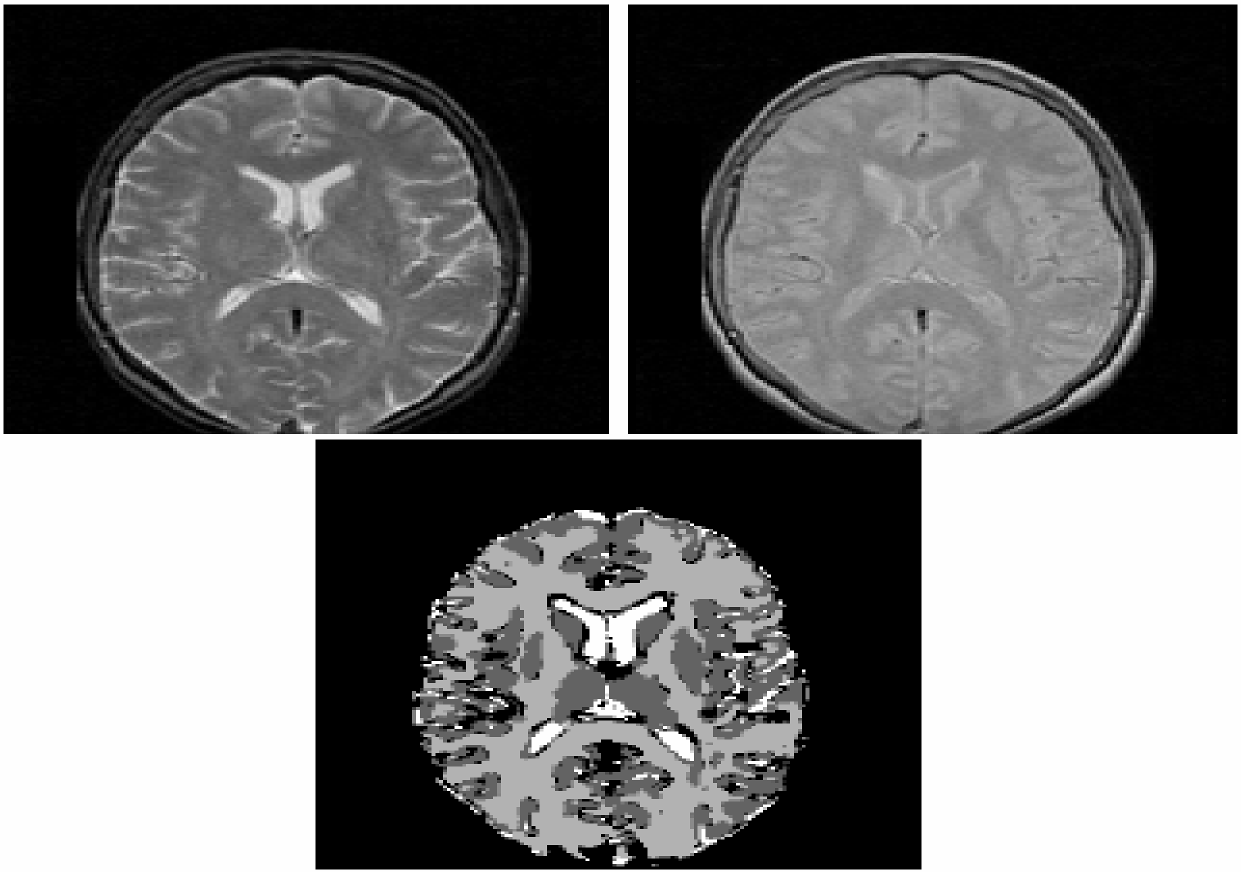
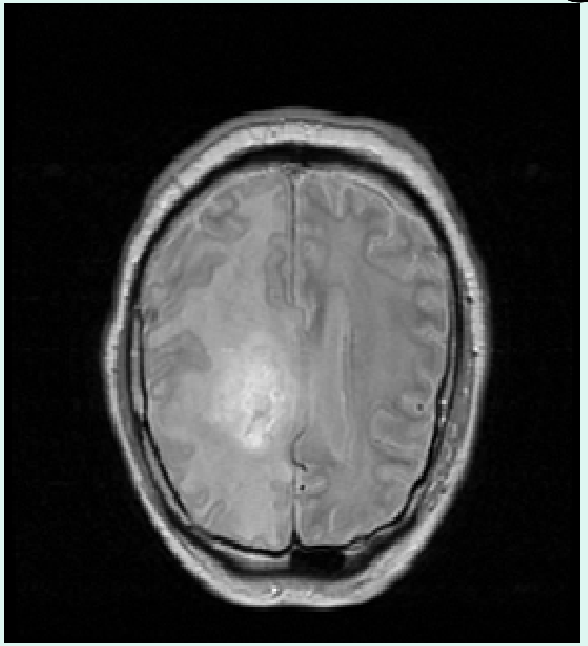
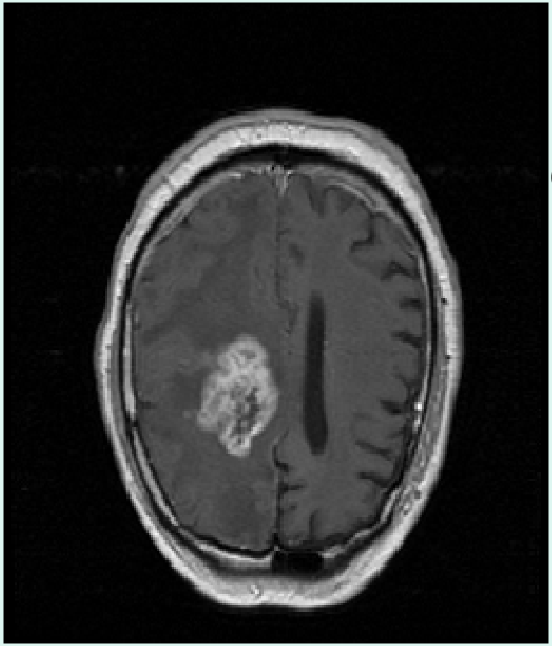


Figure 1[illegible]: Segmentation of one scan with PD and T_2 images.

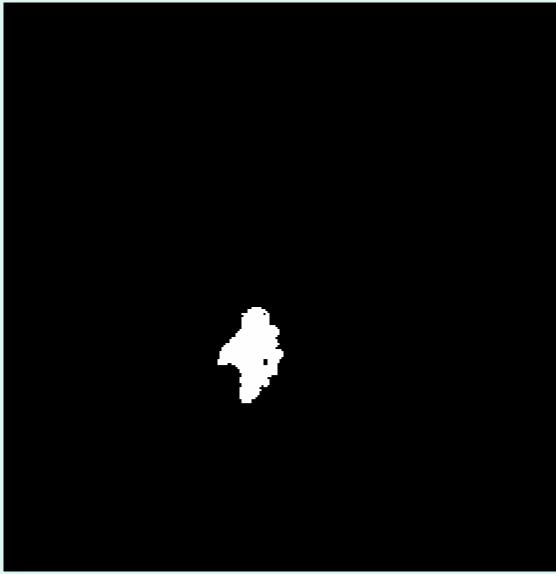
Extracting Abnormal Objects from MRI Images of the Brain





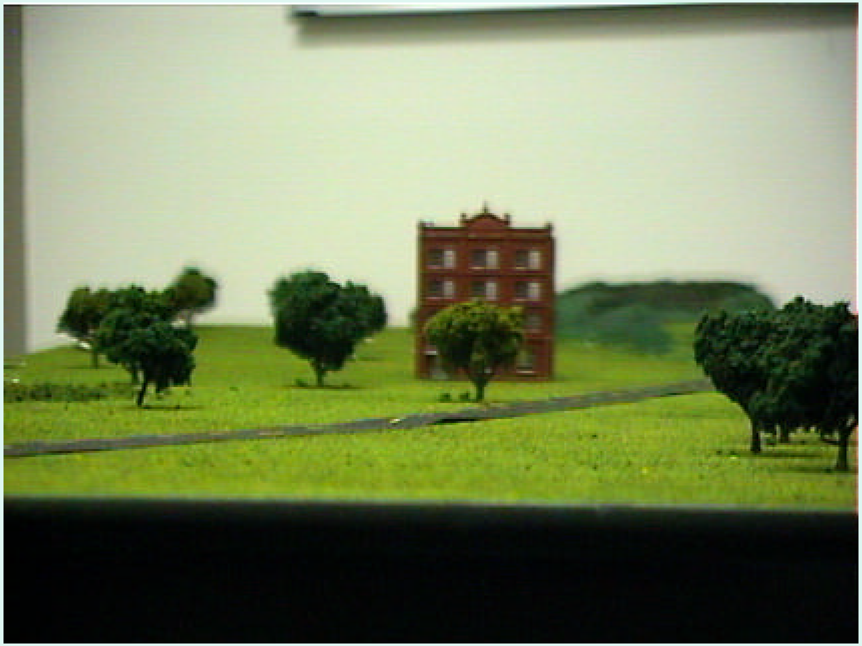
Separating Healthy Tissue from Tumor

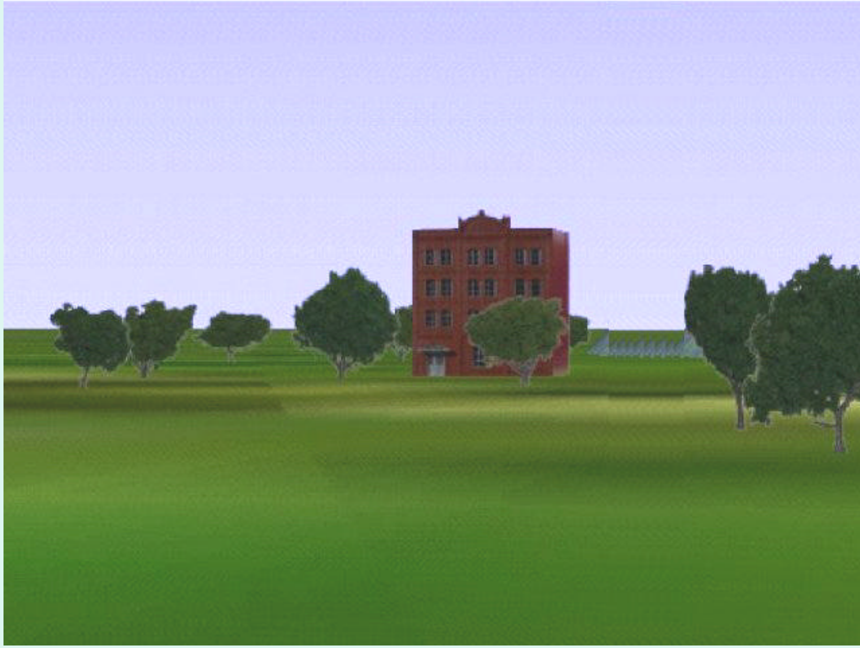
Simulating and Planning Gamma Therapy & Surgery



Extracting Tumors from MRI
T1 and T2 Images

2) Registration of Optical, IR, SAR Terrain Images (Gelenbe-Khaled 02)

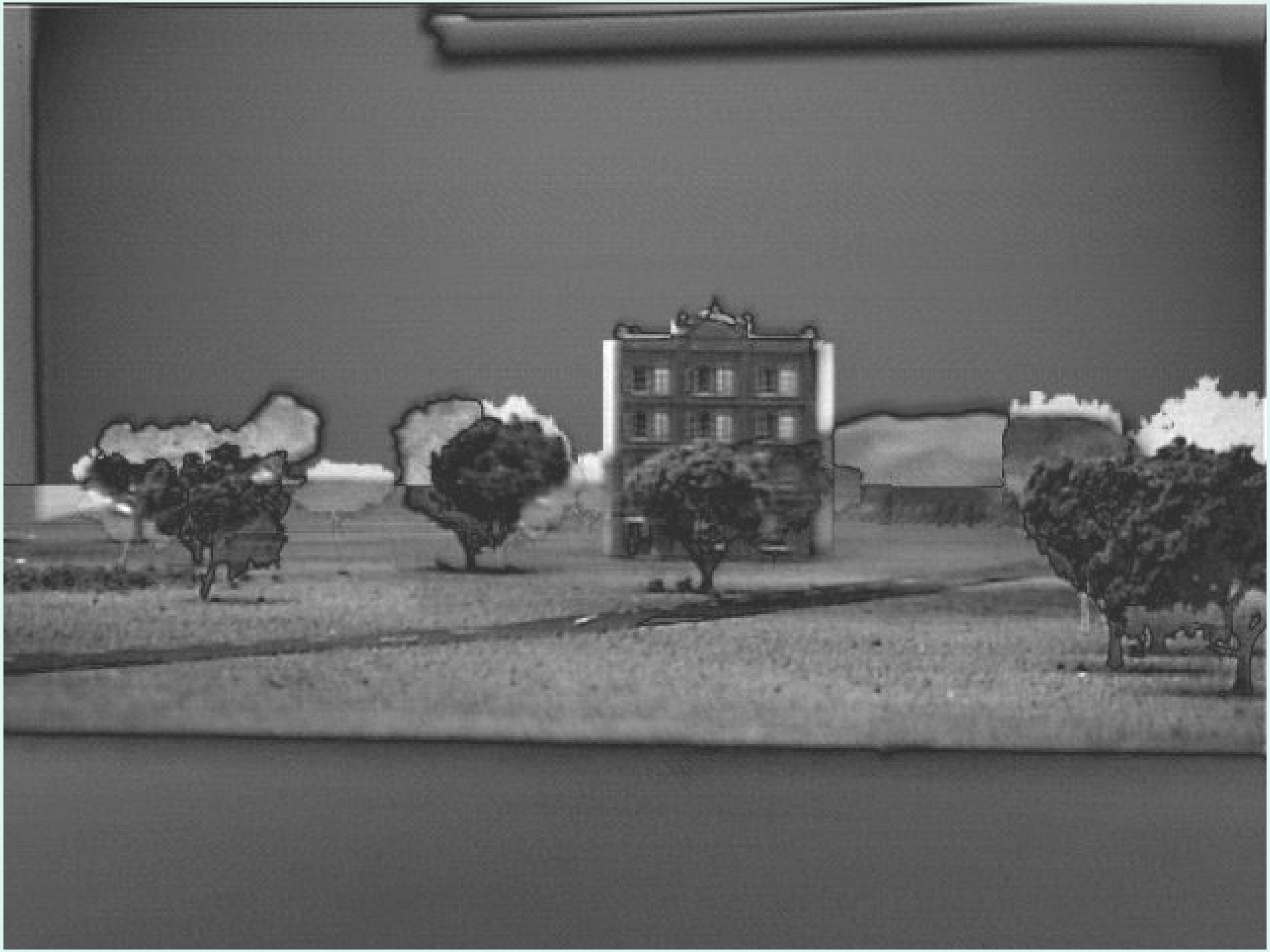




Geometric Processing of Synthetic Image to Maximise Correct Match:
Aim-point Adjustment and Image Scaling

IR Image of the same Scene

Computation of Error in Image



3) RNN based Adaptive Video Compression: Combining Motion Detection and RNN Still Image Compression

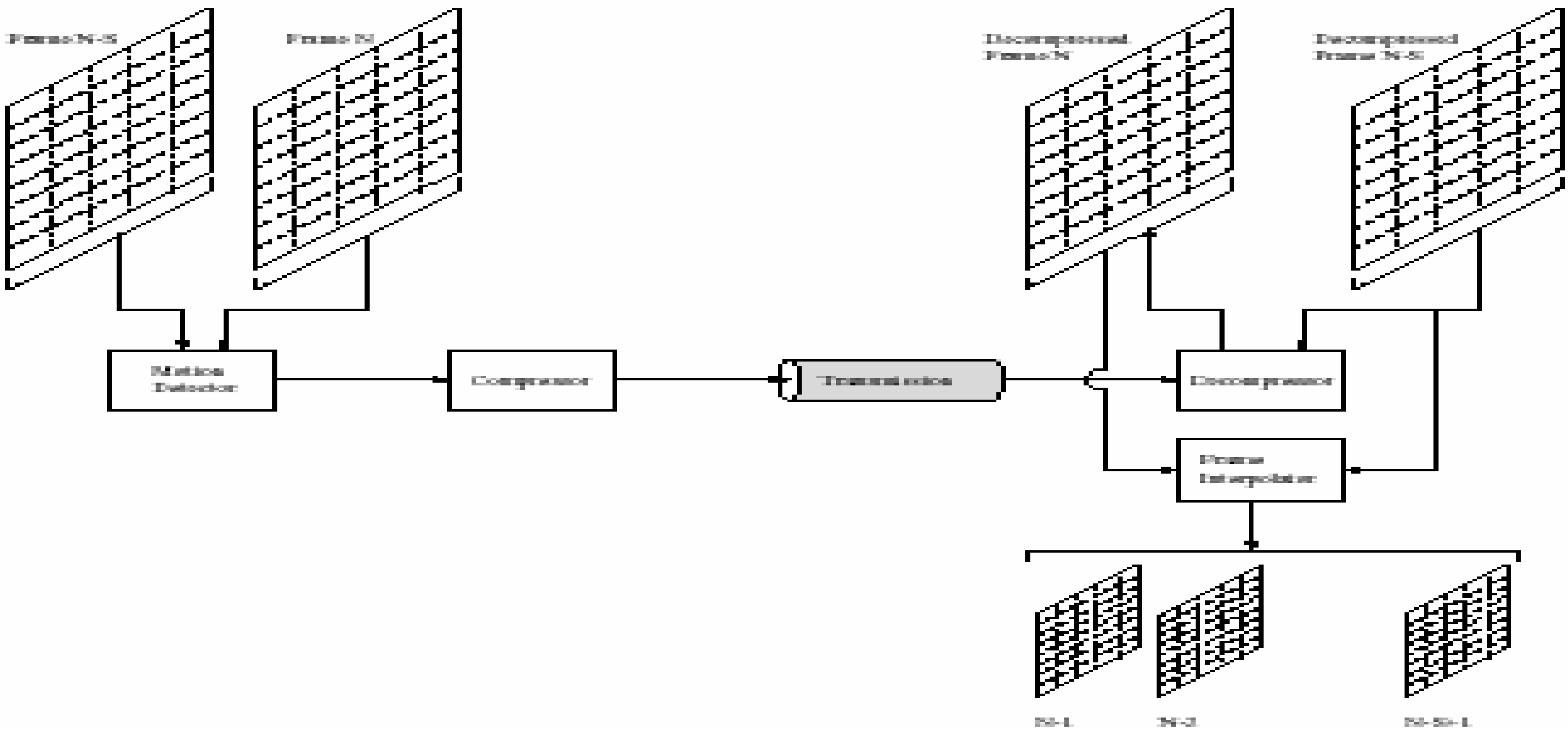


Figure [illegible]: Block diagram of the complete compression scheme.

Neural Still Image Compression
Find RNN R that Minimizes
|| R(I) - I ||
Over a Training Set of Images {I }

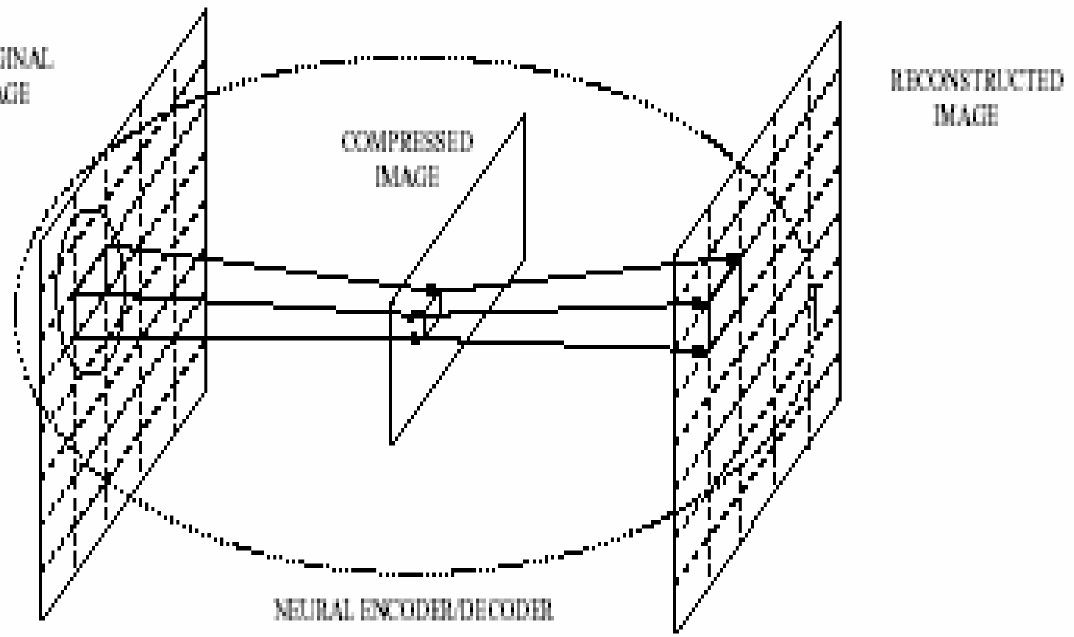


Figure 21: Compression of an arbitrarily large image using a neural encoder/decoder

RNN based Adaptive Video Compression

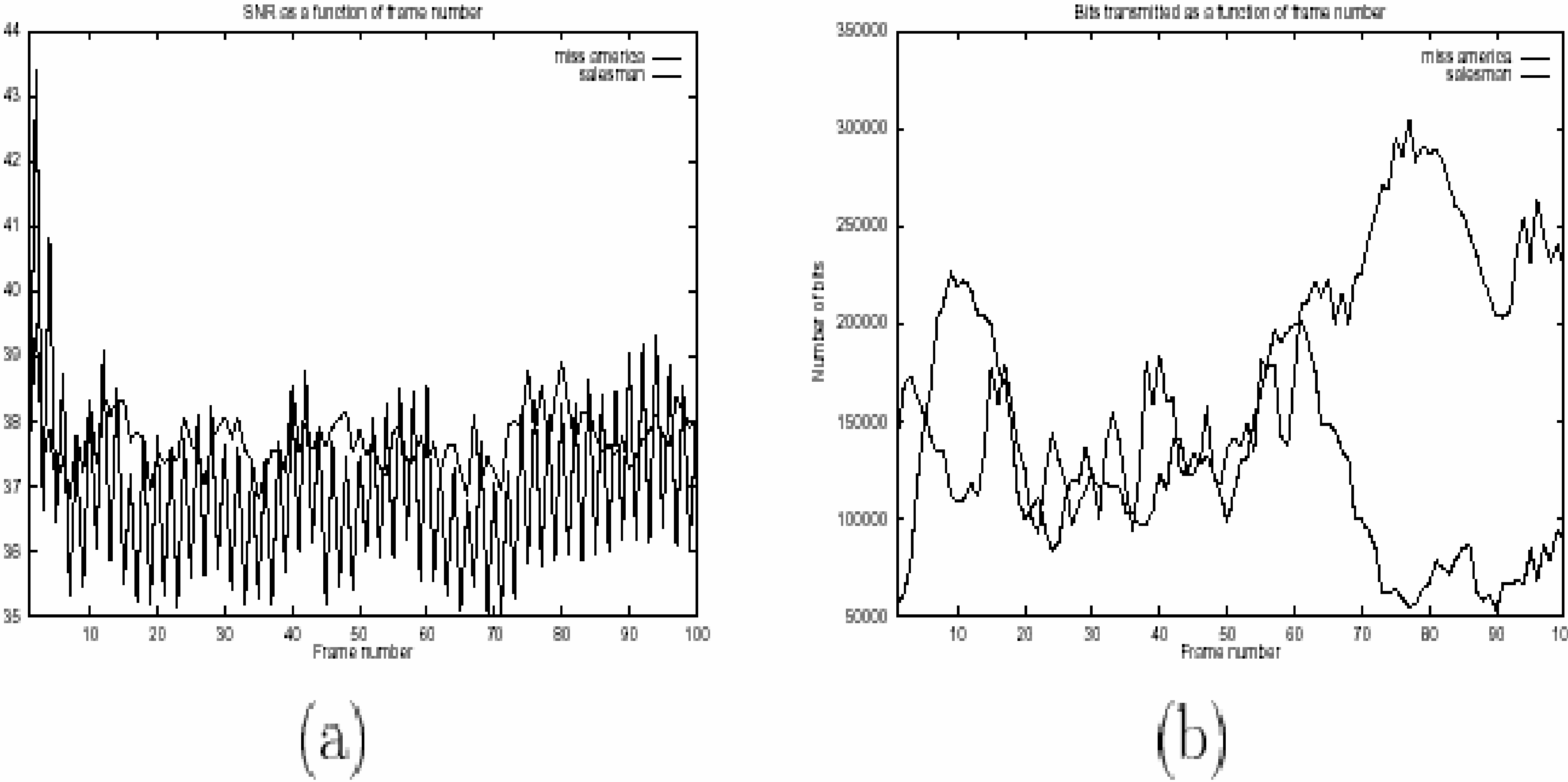
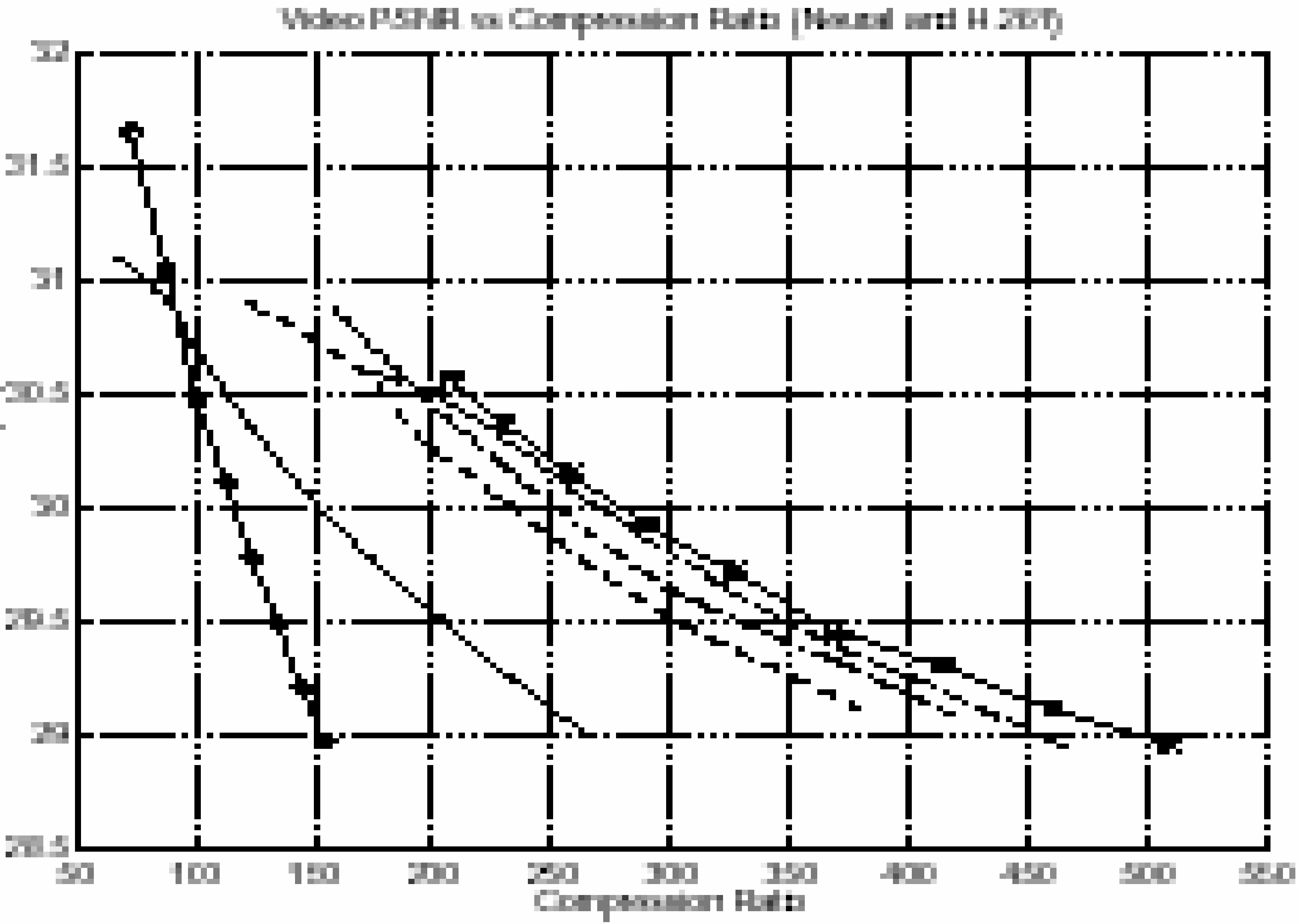


Figure 25: Experimental results for motion detection with $d = 1$: a) PSNR as a function of frame number, b) Number of bits transmitted as a function of frame number

Figure 30: Results of the 100th frame of the salesman sequence for $Q = 30$ and $d = 1$ (31.50 dB), 3 (30.99 dB), 5 (30.17 dB) and 7 (29.61 dB)


Video PSNR vs Compression Ratio (Neural and H.261)
32
31.5
31
30.5
30
29.5
29
28.5
50
100
150
200
250
300
350
400
450
500
550
Compression Ratio

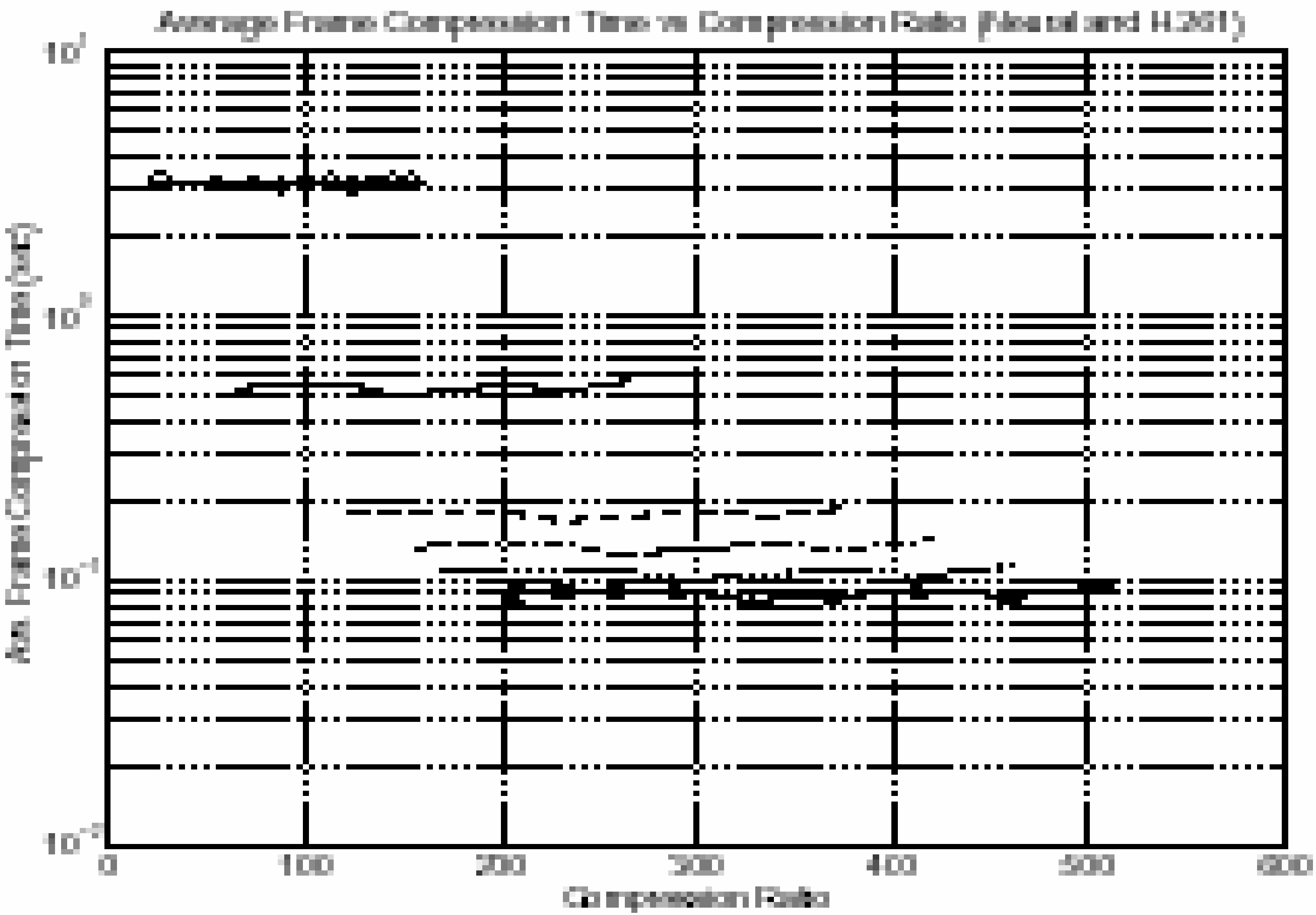


Figure 32: Compression Ratios versus Time

4) Analytical Annealing with the RNN: Multicast Routing (Similar Results with the Traveling Salesman Problem)

- Finding an optimal many-to-many communications path in a network is equivalent to finding a Minimal Steiner Tree which is NP-hard
- The best heuristics are the Average Distance Heuristic (ADH) and the Minimal Spanning Tree (MSTH) for the network graph
- RNN Analytical Annealing improves the number of optimal solutions found by ADH and MST by approximately 10%

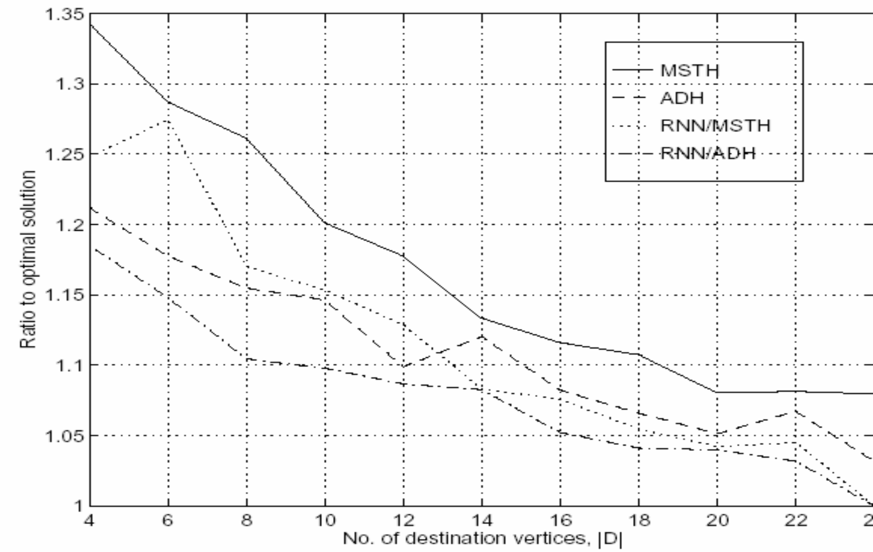


Figure 6: Worst case performance of the heuristics for graphs with 25 verti

5) Goal Based Learning

Get to destination D at minimum cost without getting lost -- If you get lost someone else will have to complete the mission

x = current position, d = position of destination

s = speed, s^{-1} = time for one step motion

D = directional decision,

$$G_D(x,d) = s^{-1} + [1-p(x+sD)]\,G\,(x+sD,d) + p(x+sD)\{G(0,x+sD) + G\,(x+sD,d)\}$$
$$= s^{-1} + G\,(x+sD,d) + 2p(x+sD)G(0,x+sD)$$

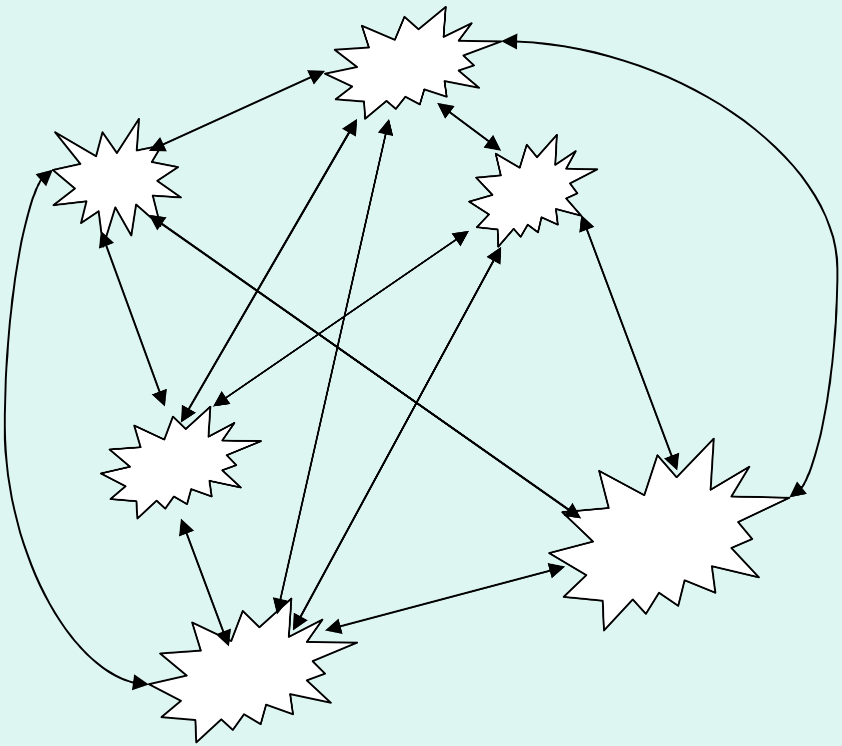
p(x+sD) = probability of getting "lost" at position x+sD

G(u,v) = cumulative cost incurred to get from point u to v

G(0,x+sD) is the cost of **bringing a new entity to x+sD**

Reinforcement Learning in G-Networks

- Fully Connected System
- Fully Recurrent with Excitatory and Inhibitory Weights
- Decision Output is the one corresponding to the largest q of all the neurons



Reinforcement Learning Algorithm

- Decision threshold – Recent Historical Value of Reward

$$T_l = aT_{l-1} + (1 - a)R_l, \; R = G^{-1}$$

- Recent Reward R_l

If

$T_{l-1} \leq R_l$ then

$$w^+(i, j) \leftarrow w^+(i, j) + R_l$$
$$w^-(i, k) \leftarrow w^-(i, k) + \frac{R_l}{n-2}, \; k \neq j$$

else

$$w^+(i, k) \leftarrow w^+(i, k) + \frac{R_l}{n-2}, \; k \neq j$$
$$w^-(i, j) \leftarrow w^-(i, j) + R_l$$

- Re-normalise all weights

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

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

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

- Compute q = $(q_1, \ldots, q_n)$ from the fixed-point
- Select Decision k such that $q_k > q_i$ for all i=1, …, n

Combining QoS-class dependent routing from G-Networks using Triggers and Reinforcement Learning – the CPN Routing Algorithm

Many Internet applications have QoS requirements.

- Voice over IP, video conferencing
- Time Critical and Secure Applications
- Network games and networked simulation
- Web based commerce and banking

– IETF has proposed QoS techniques such as IntServ, DiffServ

– In CPN Users formulate their QoS Goals, smart packets probe and make routing decisions, while dumb packets transport data and gather intelligence

In Principle, Global QoS Optimisation is Possible

Network Routing subject to QoS Constraints

Let $G = (N, L)$ be a graph with node set N and link set L. A link with origin node m and destination node n is denoted by (m, n). With $N_+(n)$ and $N_-(n)$ we denote the set of incoming and outgoing neighbors to node n, that is, respectively,

$$N_+(n) = \{m \in N : (m, n) \in L\},$$
$$N_-(n) = \{m \in N : (n, m) \in L\}.$$

With each link $l = (m, n)$, $m, n \in N$ there is an associated cost $c_{mn} \geq 0$ and delay $d_{mn} \geq 0$. If $p = (m_1, \ldots, m_k)$ is a directed path (a subgraph of G consisting of nodes $m_1, \ldots, m_k$, $m_i \neq m_j$ for all $1 \leq i, j \leq k$, $i \neq j$, and links (m_i, m_{i+1}), $1 \leq i \leq k - 1$) then we define the cost and delay of the path respectively,

$$C(p) = \sum_{(m,n) \in p} c_{mn},$$
$$D(p) = \sum_{(m,n) \in p} d_{mn}.$$

The set of all paths with origin node s, destination node n and delay less than of equal to d is denoted by $P_{sn}(d)$. The set of all paths from s to n is denoted simply by P_{sn}. For any d, we are interested in finding a path $p^* \in P_{sn}(d)$ such that

$$C(p^*) \leq C(p) \text{ for all } p \in P_{sn}(d).$$

Difficulties of Global Optimisation

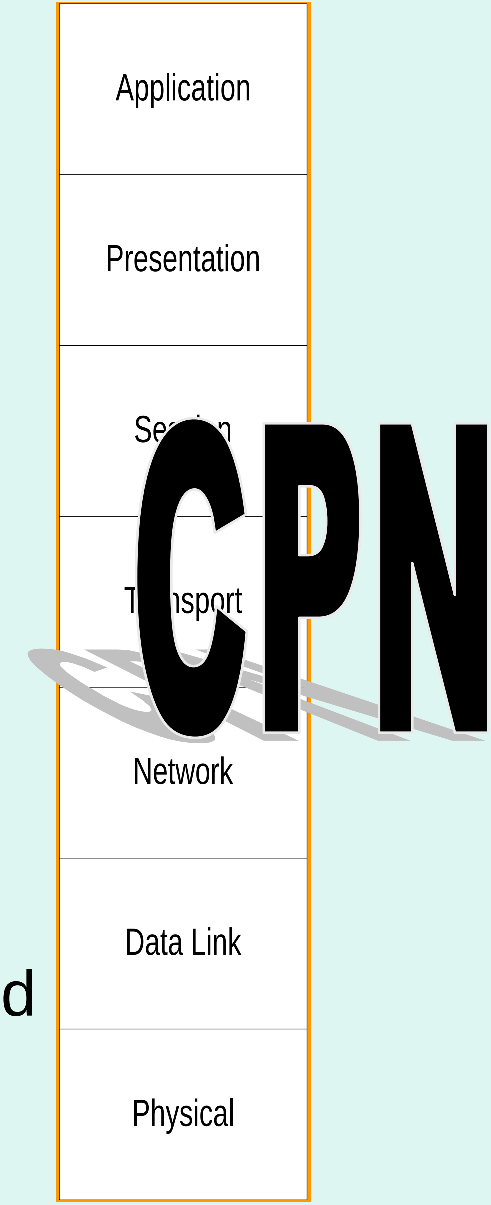
- The network is very large – for specific users, optimisation is relevant for a subset of routes at a time
- The system is large .. information delay, control delay and combinatorial explosion: global algorithms can be very slow and come too late
- The system is highly dynamic – traffic varies significantly over short periods of time
- There are large quantities of traffic in the pipes – congestion can occur suddenly, reaction and detours must be very rapid
- Measurements local to subset of users, and adaptivity is needed which is relevant to the users most concerned by the measurements

CPN Philosophy

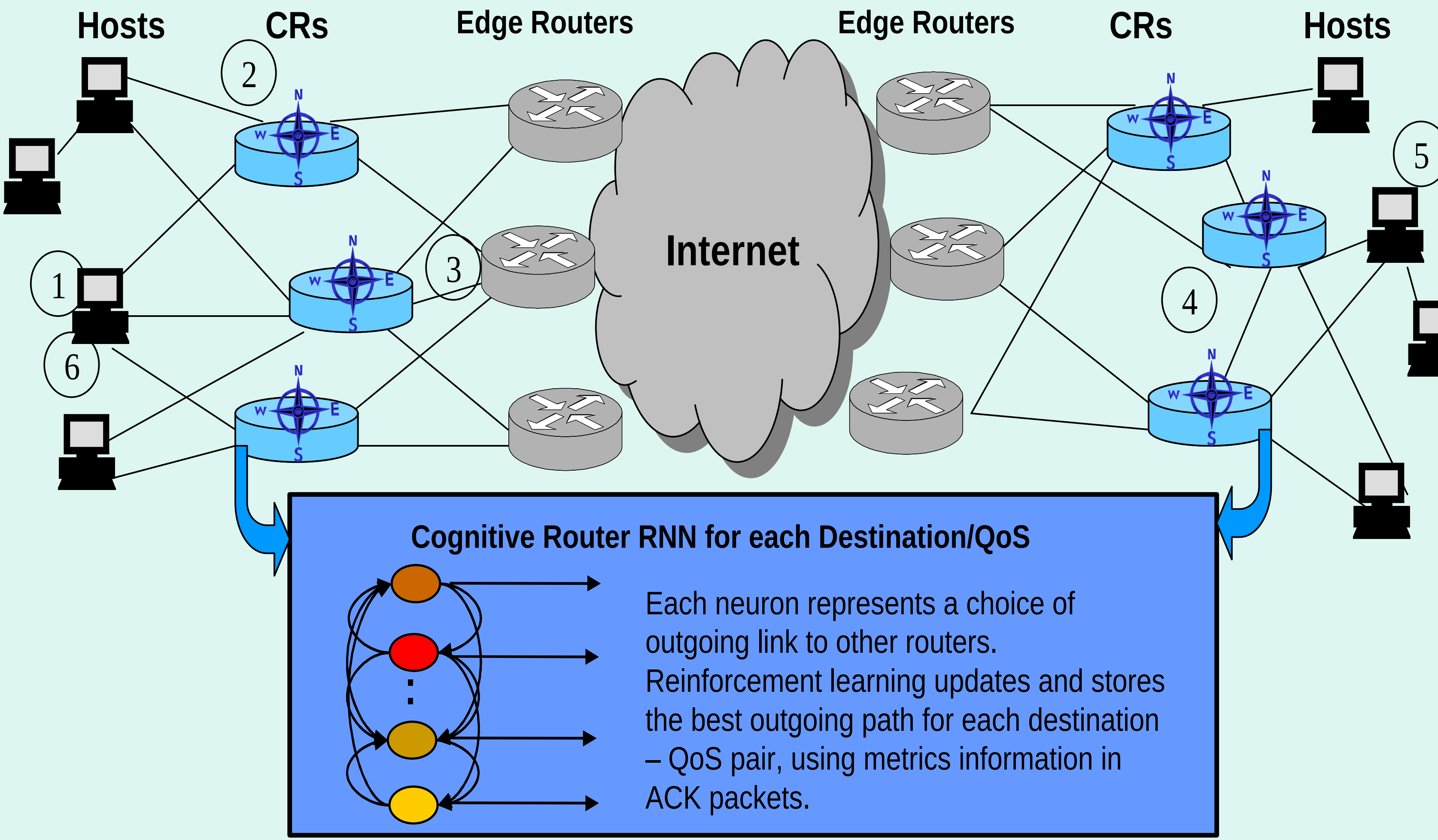
- Let the Measurements and the Adaptivity be under user control
- Let the user make his/her own QoS and economic decisions
- Remain close to, and compatible with IP

OSI Layers & CPN

- TCP/IP is a layered protocol stack
- *Application* handles particular applications, *Presentation* handles compression and encryption of data, *Session* controls establishment, management, termination of sessions
- *Transport* provides flow of data
- *Network* handles the transmission of packets in the network
- *Data-link* is responsible for the interaction of the device driver in the operating system and the network card in the machine
- *Physical* defines electrical and mechanical specifications



QoS in the Internet can be Controlled via CPN's Cognitive Routers Operating at the Periphery of the IP World



Cognitive Router RNN for each Destination/QoS

Each neuron represents a choice of outgoing link to other routers. Reinforcement learning updates and stores the best outgoing path for each destination – QoS pair, using metrics information in ACK packets.

Quote from LM & Cisco's Proposal to Darpa

March 8, 2004

" … We believe the next evolution in path switching will be the employment of various infrastructure sensing devices to collect path data and allow a user, a network administrator, or an automated process to specify paths to routers within their domain authority … … the cognitive router (CR) schema that was developed by Erol Gelenbe under the rubric of "cognitive packet networks" (CPN) [2] … represents a dramatic change in the ability of a network to make intelligent routing decisions. CRs use neural networks that essentially form multi-dimensional routing tables that respond immediately to the route performance parameters captured by the packets flowing through them. CRs employ extended QoS parameters and can change routes when they recognize route degradation. Because decisions are based only on local information provided by the smart packets, CRs are not afflicted with the problems inherent in … BGP … "

CPN Principles

- CPN operates seamlessly with IP and creates a self-aware network environment
- Users assign goals
- Packets collectively learn to achieve the goals
- Learning is performed by sharing information between packets
- Packets sharing the same goals can be grouped into *classes*
- Nodes are storage centers, mailboxes and processing units

CPN and Smart Packets

Smart Packets route themselves based on QoS Goals, e.g.,

- Minimise Delay or Loss or Combination
- Minimise Jitter (for Voice)
- Maximise Dispersion (for security)
- Minimise Cost
- Optimise Cost/Benefit

Smart Packets make observations & take decisions

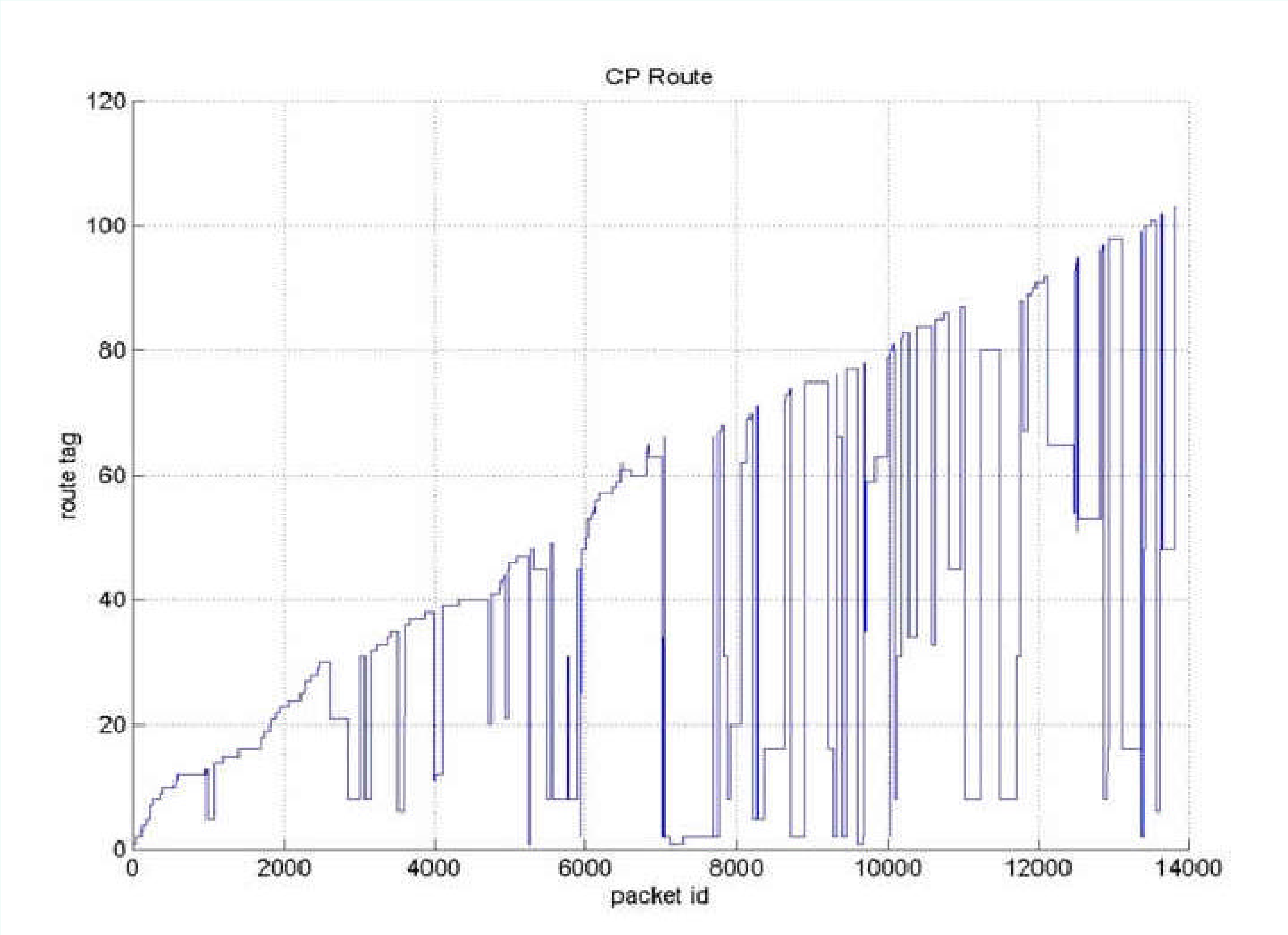
ACK Packets bring back observed data and trace activity

Dumb Packets execute instructions, carry payload and also may make observations

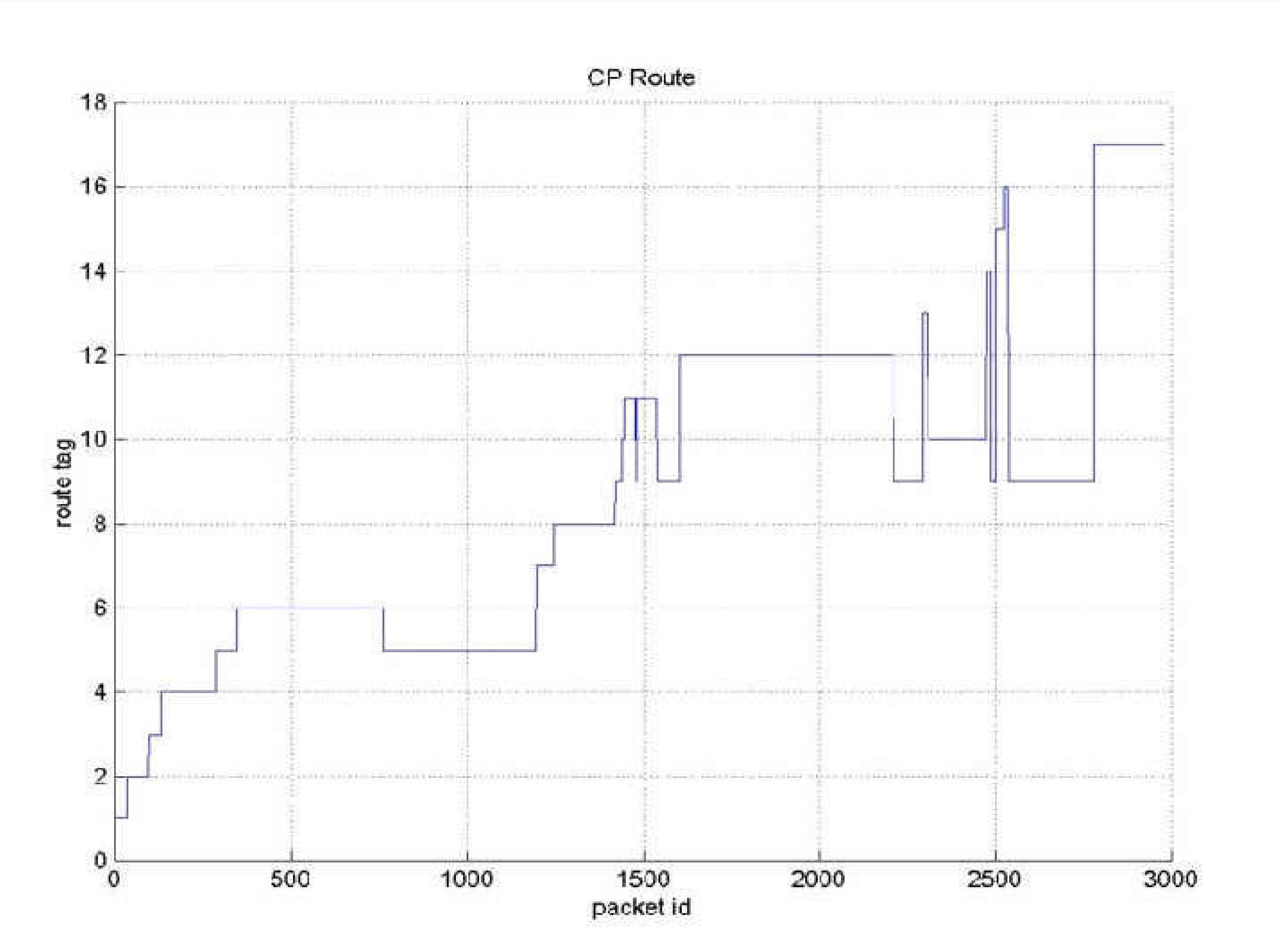
Cognitive Adaptive Routing

- Conventional QoS Goals are extrapolated from Paths, Traffic, Delay & Loss Information – this is the “Sufficient Level of Information” for Self-Aware Networking
- Smart packets collect path information and dates
- ACK packets return Path, Delay & Loss Information and deposit W(K,c,n,D), L(K,c,n,D) at Node c on the return path, entering from Node n in Class K
- Smart packets use W(K,c,n,D) and L(K,c,n,D) for decision making (e.g. Reinforcement Learning)

CPN Test-Bed Measurements
Ongoing Route Discovery by Smart Packets



CPN Test-Bed Measurements
Ongoing Route Discovery by Smart Packets



QoS Driven Application
Voice over CPN

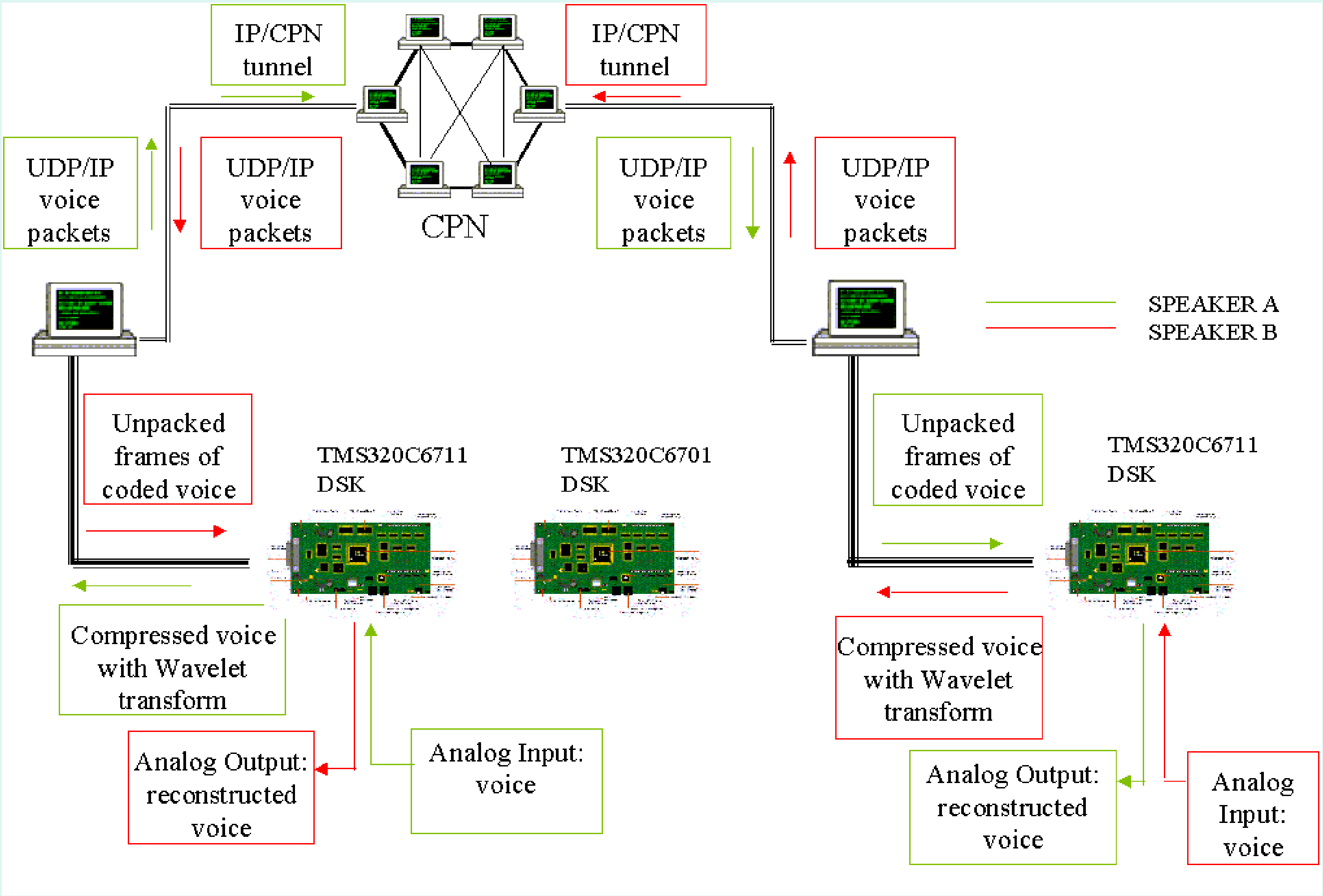


Fig. 1. Voice over CPN

Experimental Results
Voice over CPN

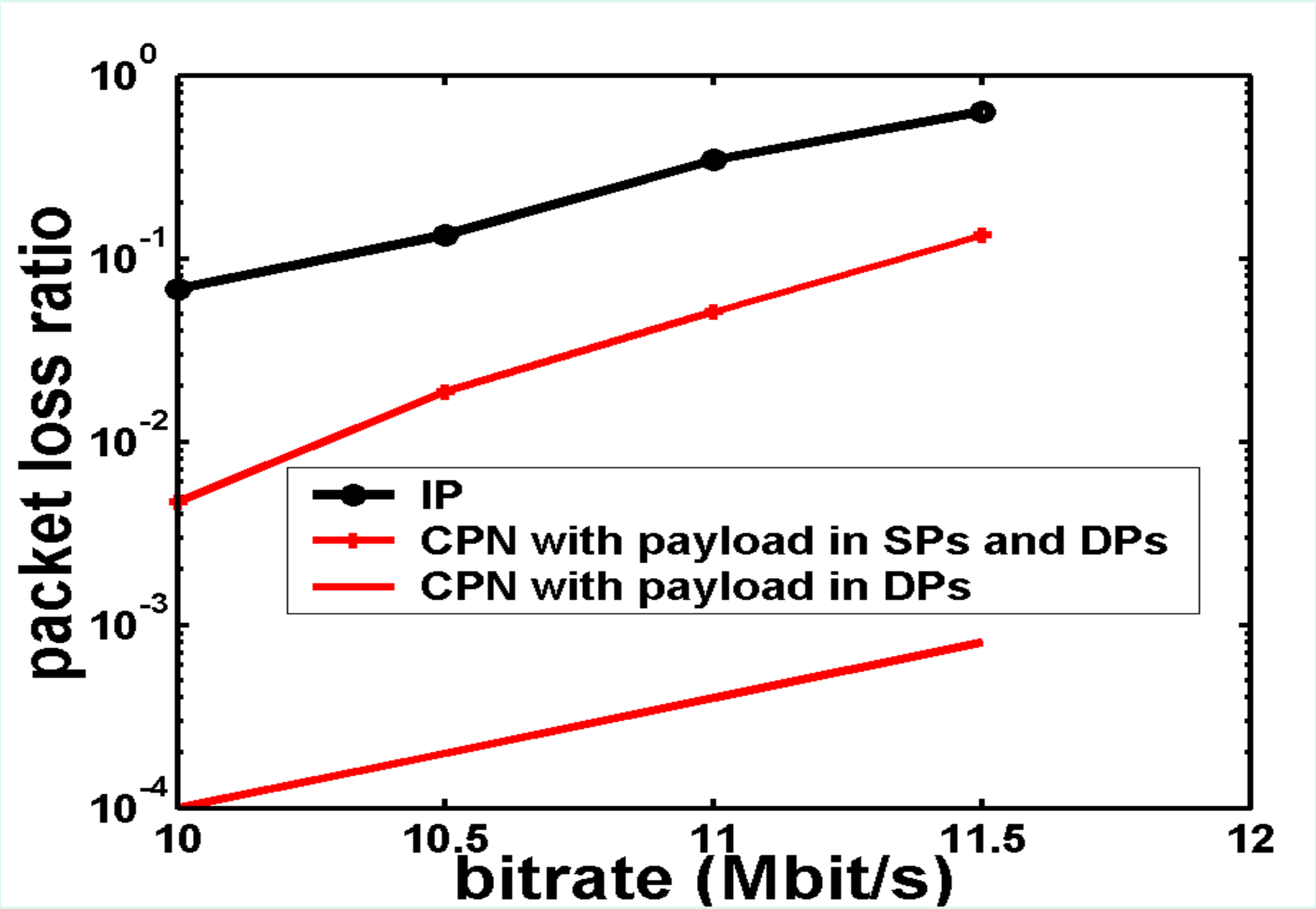
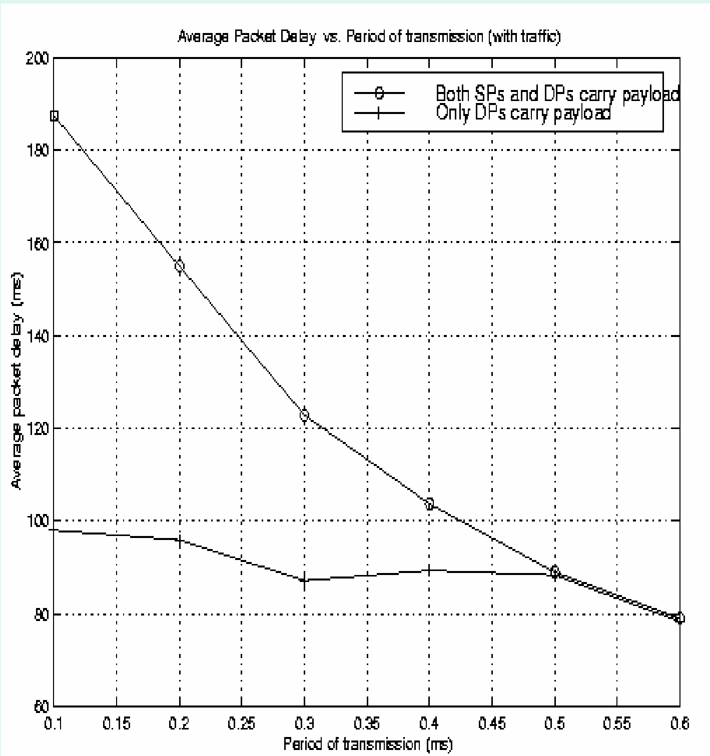


Fig. 4

Experimental Results
Voice over CPN



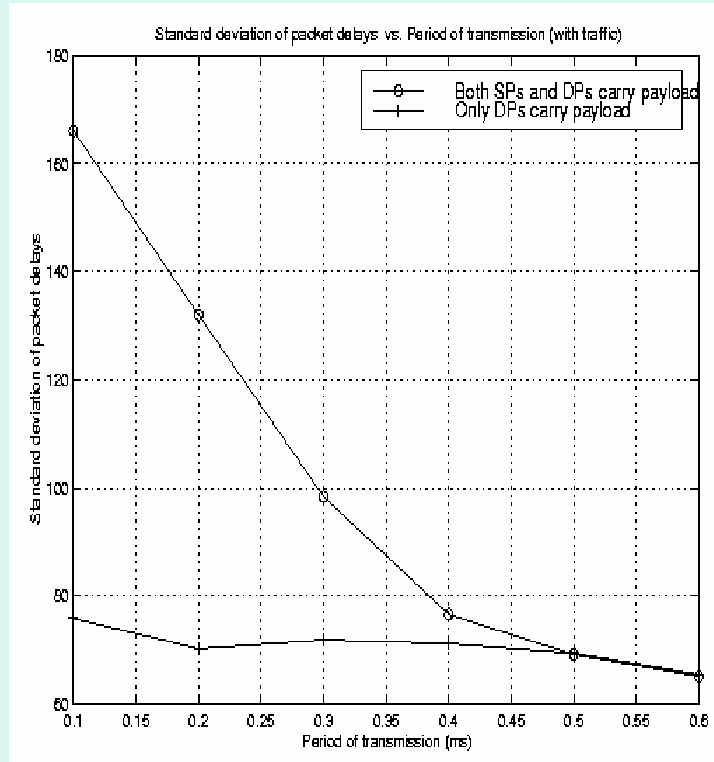


Fig. 6 : **Average round-trip delay (left) and jitter (right) for user payload when only DPs are allowed to carry user payload**

Experimental Results
Voice over CPN

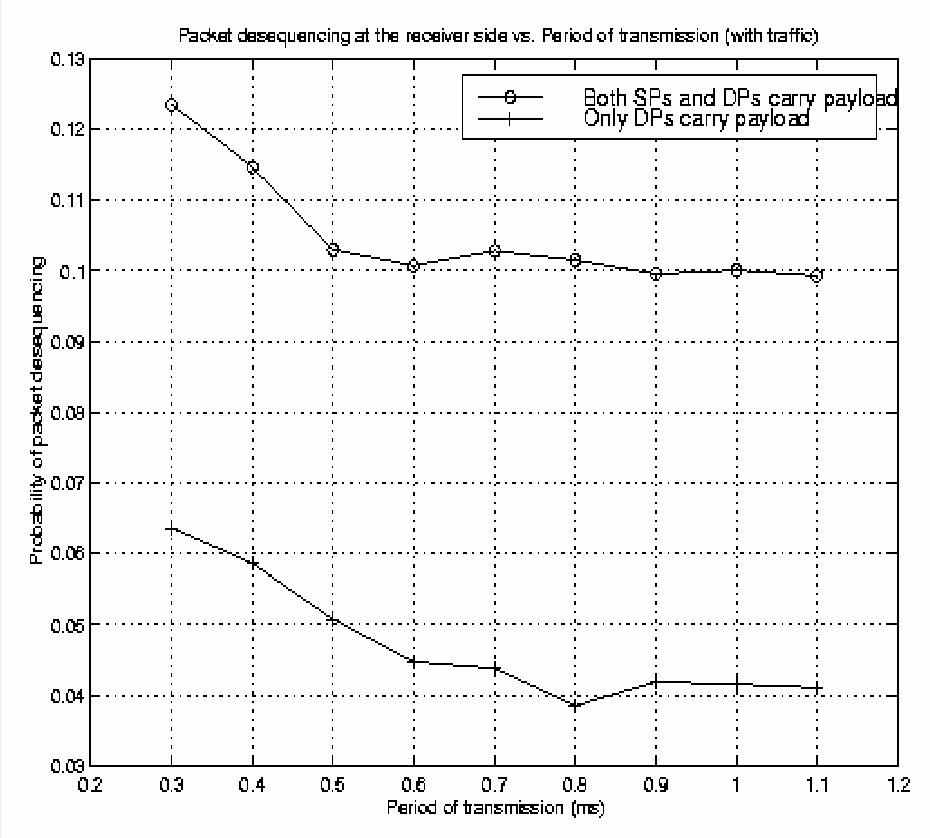
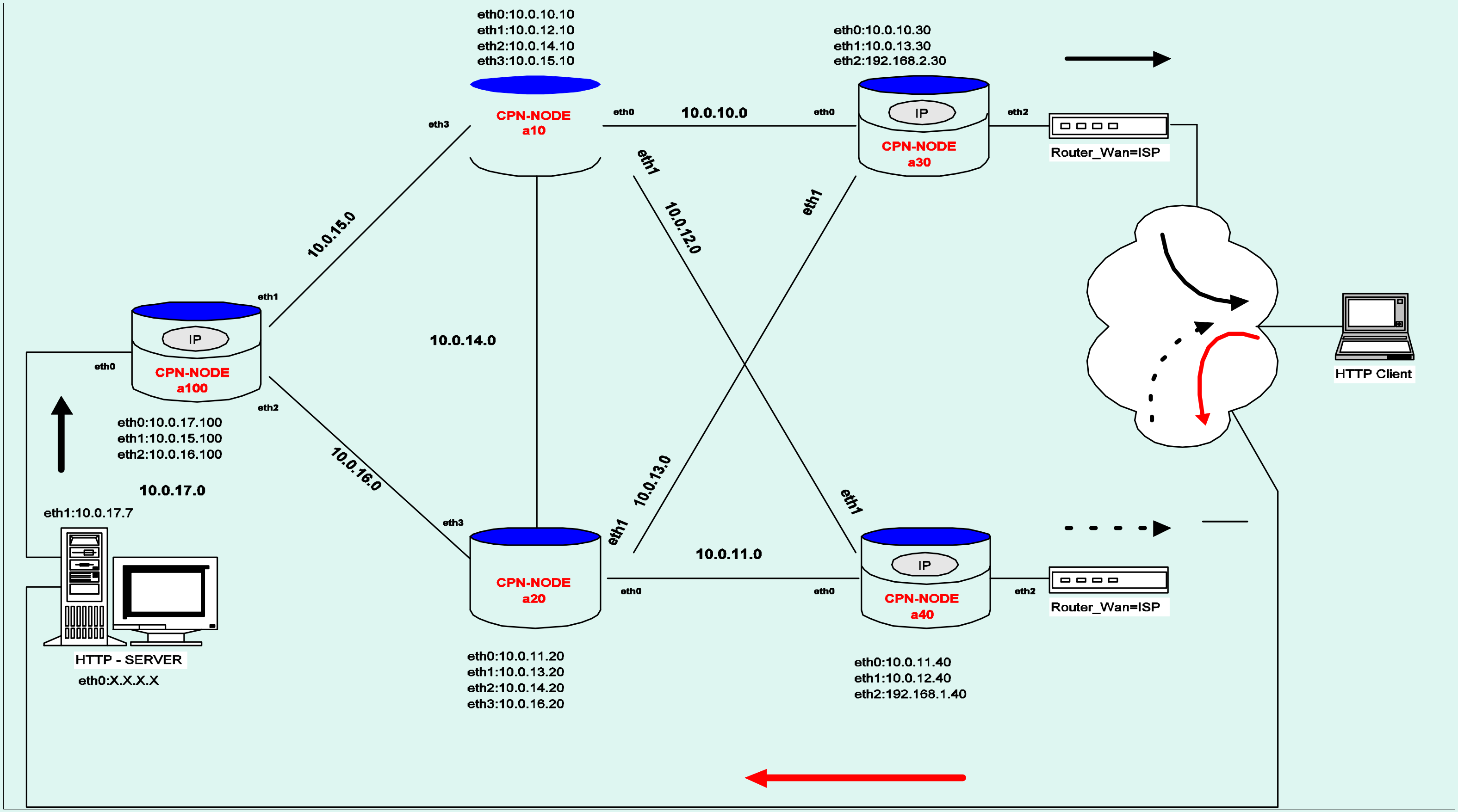
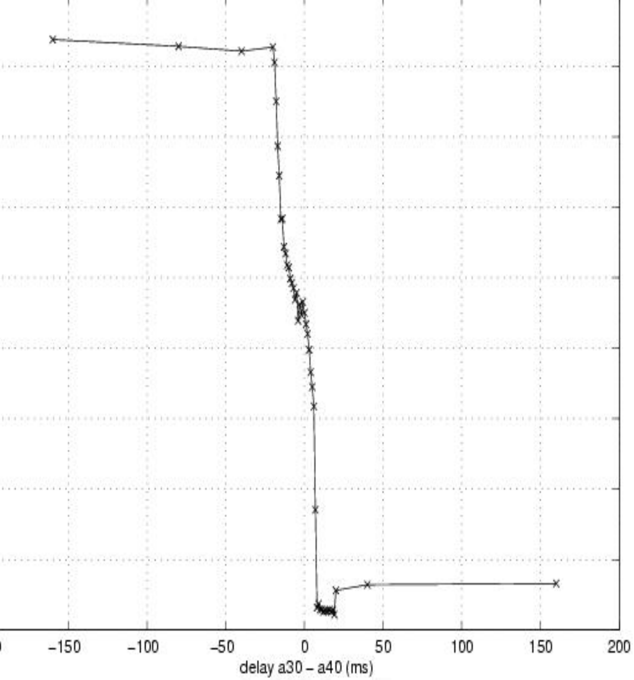
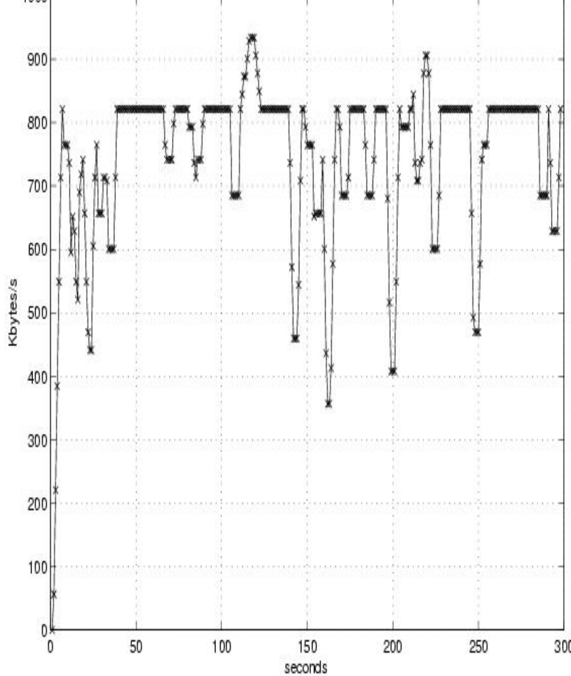


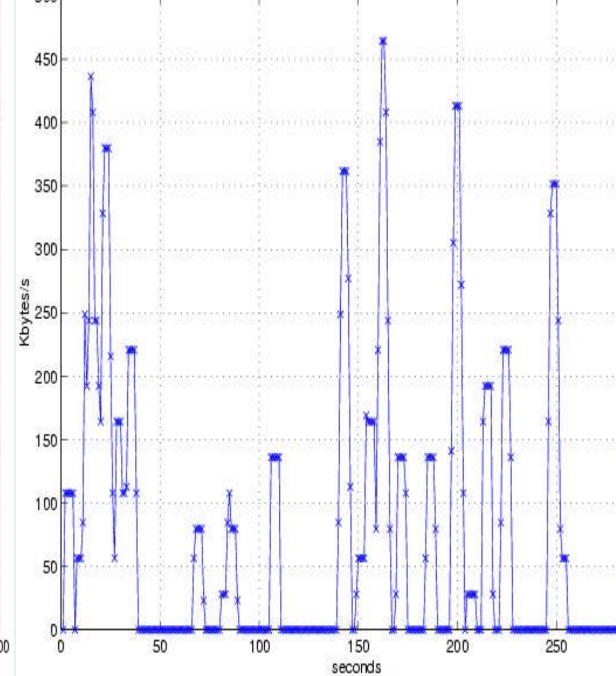
Fig. 7. Probability of packet desequencing perceived by the receiver side

CPN for Traffic Engineering

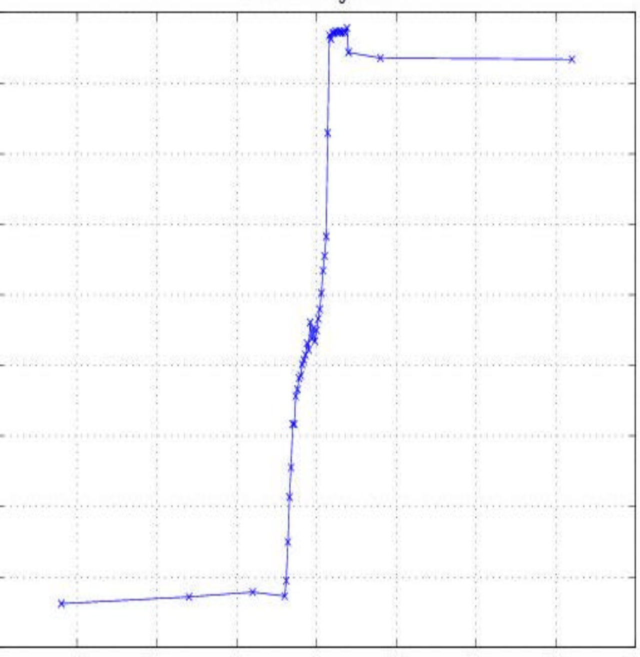




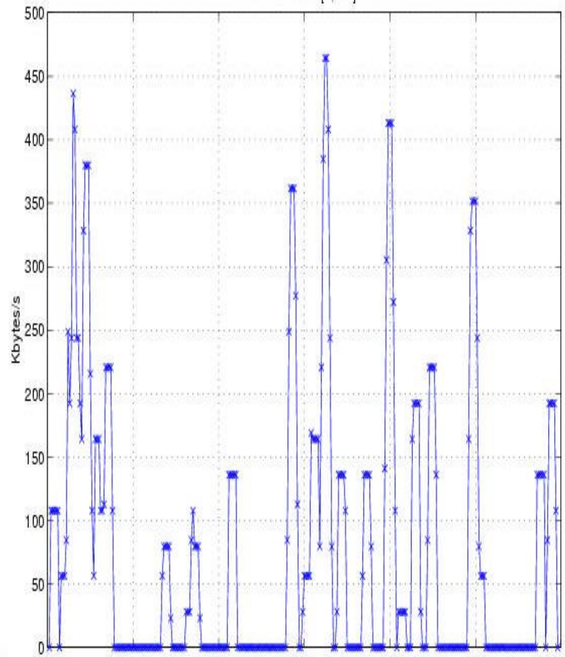


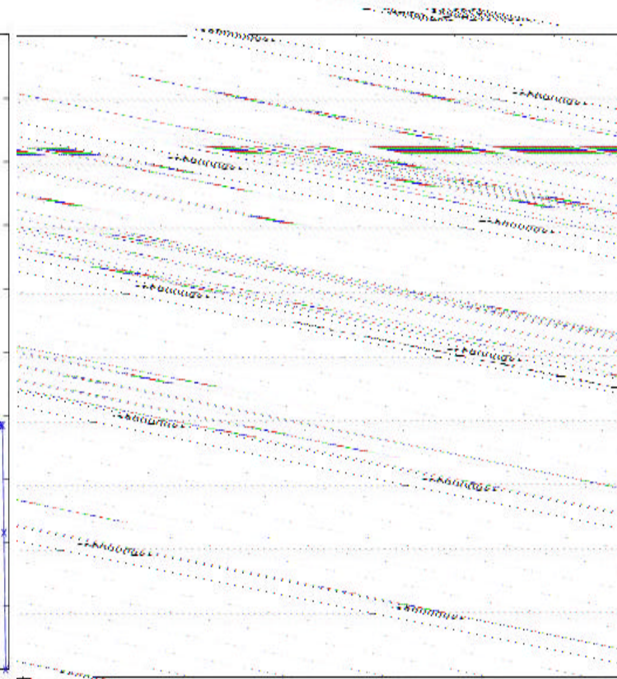


Load Balancing – a40

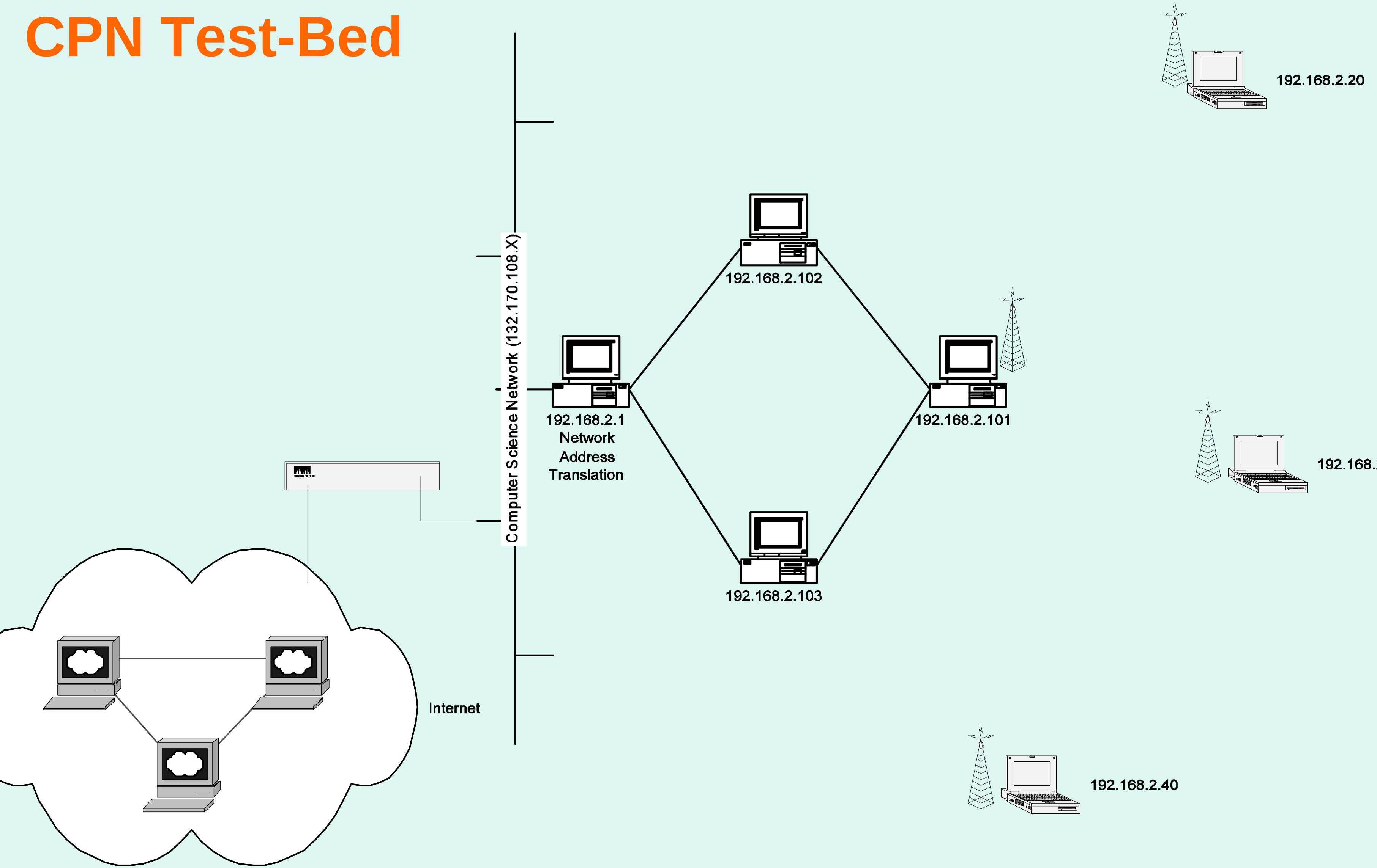


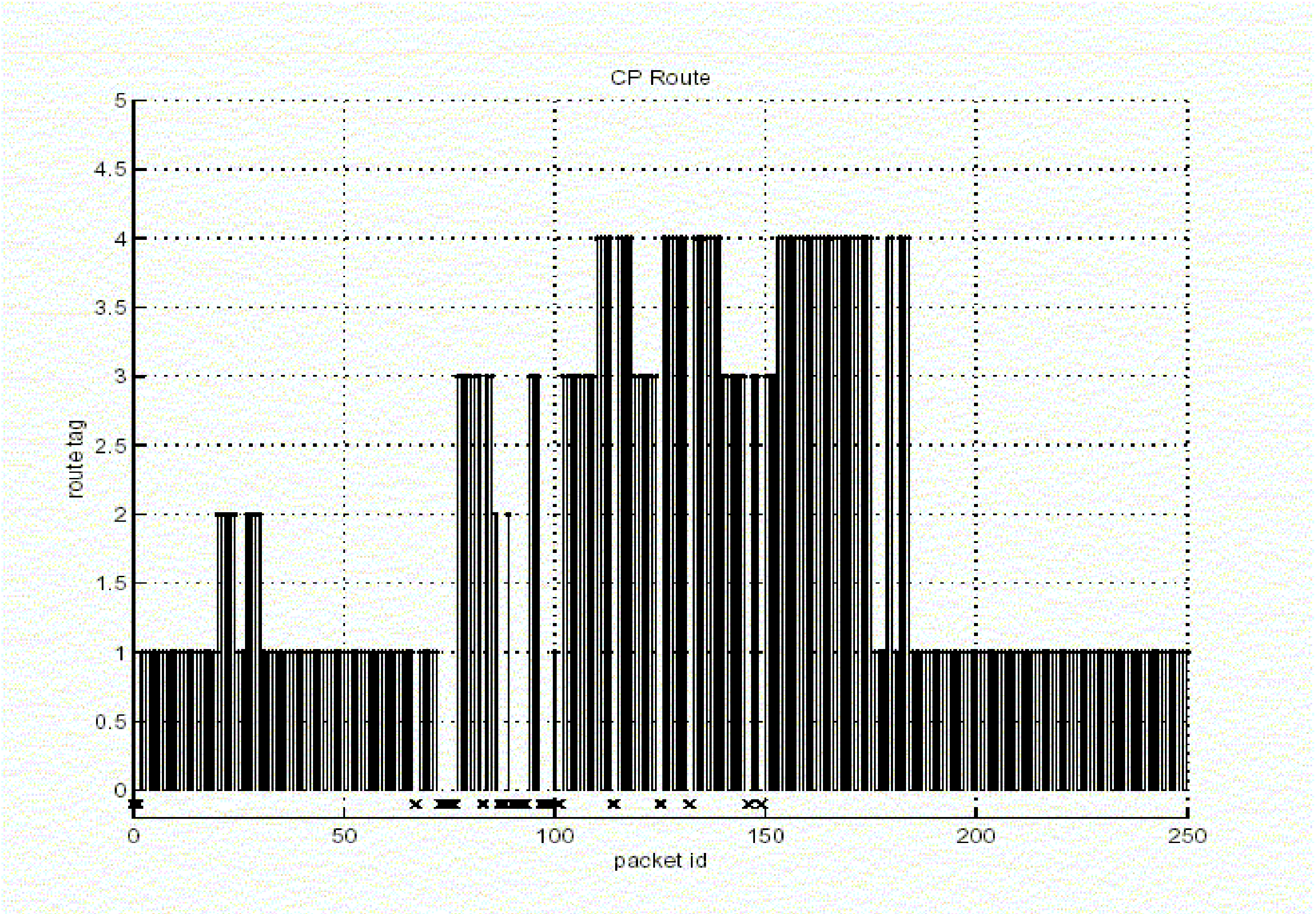
Traffic – a40 [0,160]

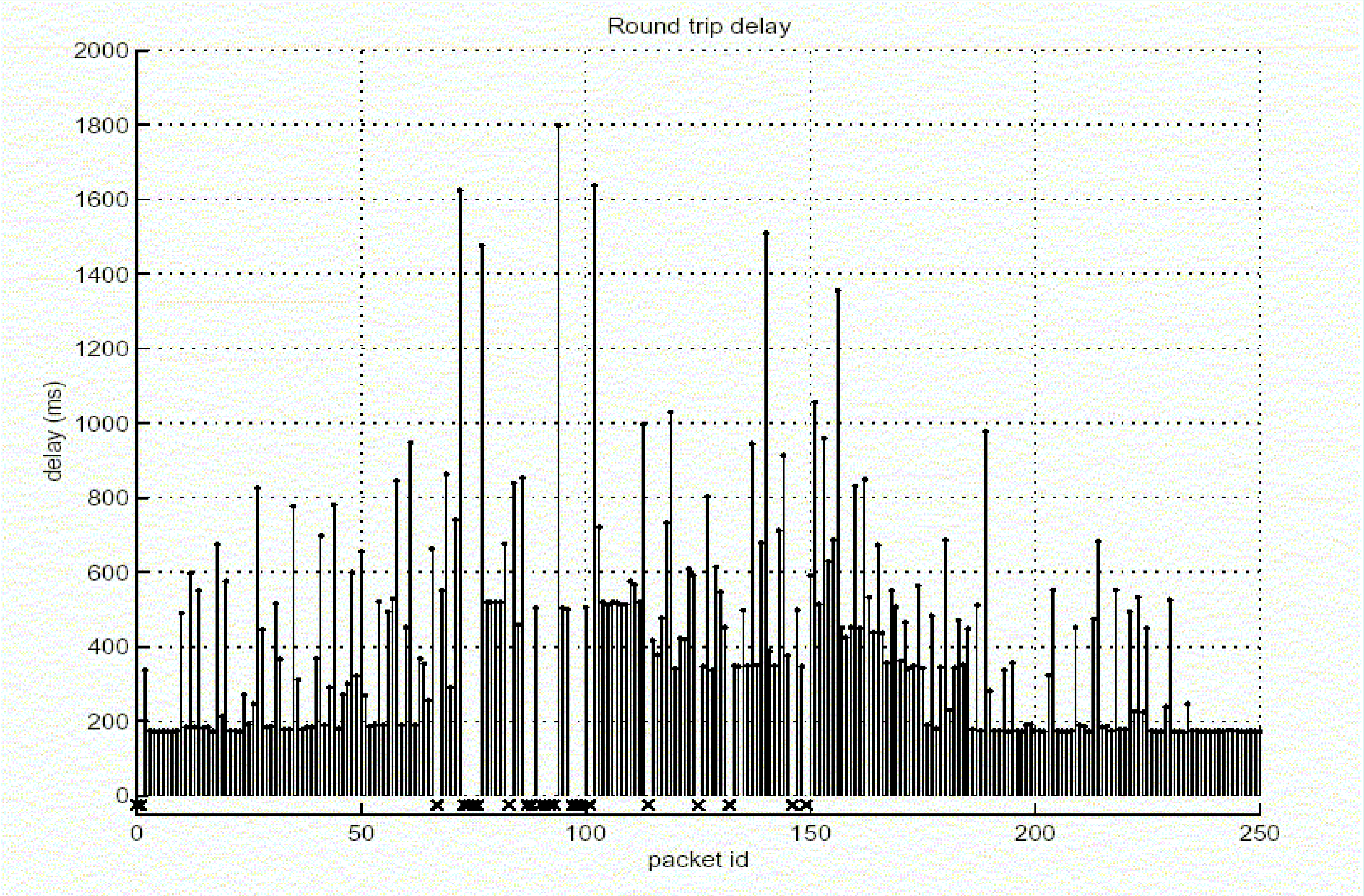




Wireless Power-Aware Adhoc CPN Test-Bed




CP Route
route tag
packet id
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
0
50
100
150
200
250


Round trip delay
delay (ms)
packet id
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0
50
100
150
200
250

… as we conclude …

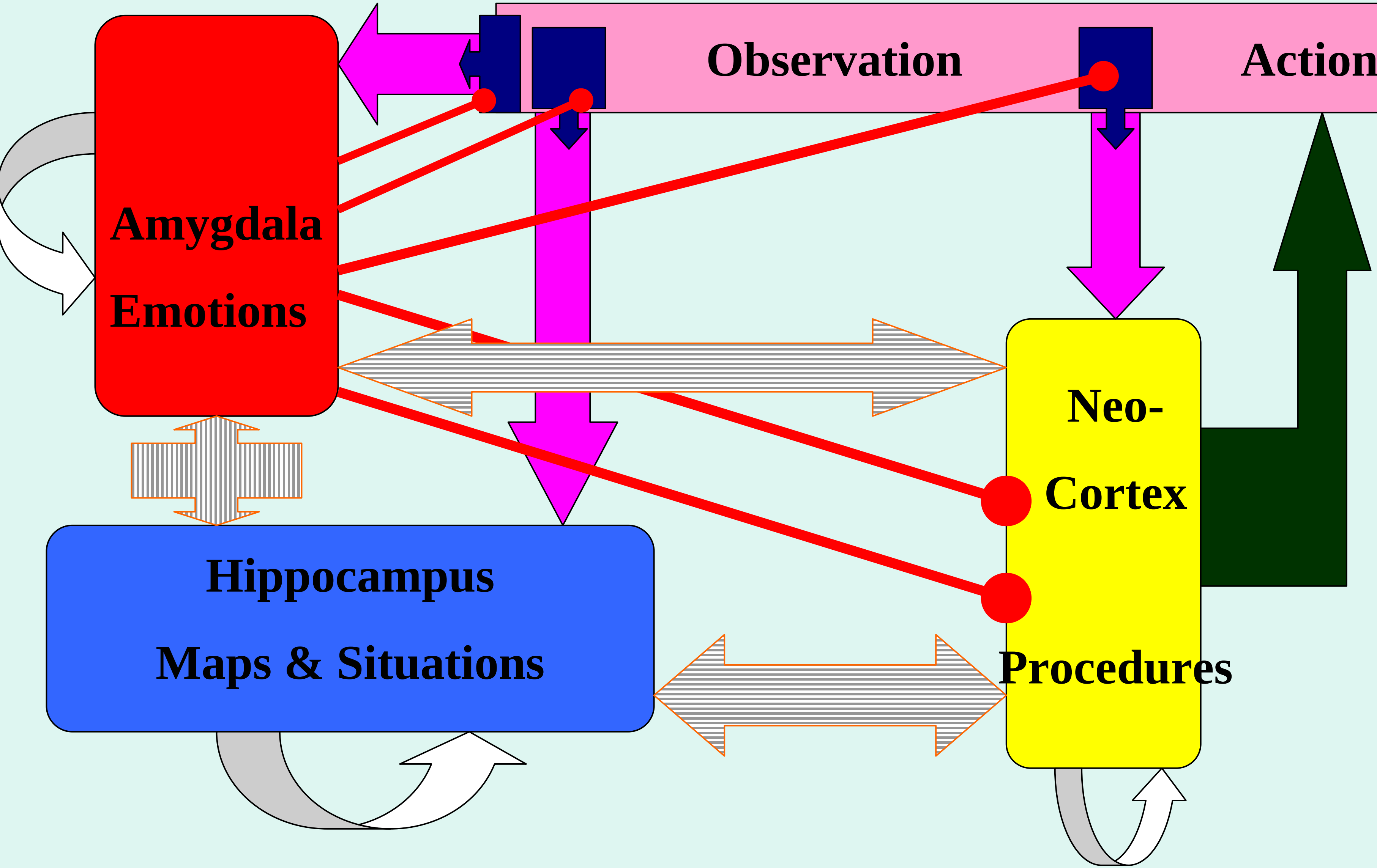
- The challenges for the next decade are:

  - To model accurately increasingly complex natural brain systems and understand their functional behaviour

  - To exploit neural paradigms in order to

  provide adaptive control for complex socio-economic systems (health systems, business systems, trading systems, IT networks ..)

Model accurately complex natural brain systems and understand their functional behaviour



Neural Network Control of Socio-Economic Systems: e.g. DARPA's Cognitive Networks