

FRAME - Fixed Route Adapted Media Streaming Enhanced Handover Algorithm

Enda Fallon, Liam Murphy, *Member, IEEE*, John Murphy, *Senior Member, IEEE*, Gabriel Miro-Muntean *Member, IEEE*

Abstract—One of the key features of the Media Independent Handover (MIH) framework, introduced by the IEEE 802.21 standard, is the support for events, including network degradation events which can be triggered based on link layer metrics and propagated to upper layer mobility protocols. As a framework, MIH does not provide specifics on how these events are triggered. Typically events are triggered when performance parameters such as Received Signal Strength (RSS) and link loss rate exceed a predefined threshold. In this paper we suggest that for vehicular systems, the constrained nature of movement enables network performance prediction. We propose to capture this performance predictability through a Fixed Route Adapted Media-streaming Enhanced handover algorithm (FRAME). FRAME uses a directed feed forward neural network to trigger MIH link events. FRAME provides a pluggable learning mechanism which allows for the extensible definition of performance and learning metrics. FRAME is evaluated using a commercial metropolitan network implementation. Results show that FRAME has significant performance improvements over existing MIH link triggering mechanisms.

Index Terms—heterogeneous networking, media streaming, vehicular networks, MIH, directed learning

I. INTRODUCTION

Having promised much in the late 1990's early 2000's, IP enabled wireless and mobile networks are finally realising their potential. Technologies such as UMTS (3G) were generational in nature and proposed to support the mobile computing requirements of the "dot com" era. A decade later, mobile devices such as the iPhone, iPad and Android smart phones mean IP enabled access networks are receiving the level of utilisation originally anticipated.

Wireless LAN (WLAN) was originally designed to provide coverage in specific "hot spot" areas. The advent of heterogeneous networking has enabled WLAN to play a significant role as a constituent part of a wider IP access network infrastructure. The significant numbers of WLAN installations providing high capacity low cost network access make it the network of choice for end users. When WLAN coverage is not available, users migrate to a metropolitan or mobile access network.

In such a scenario an effective network migration strategy is required to moderate between the physical characteristics of the underlying network and the QoS required by the target application. The IEEE, through the 802.21 working group, have proposed the MIH standard [1][2][3]. As a framework MIH provides the concept of communication of network

critical events to upper layer mobility protocols. While MIH defines the communication interface, it does not provide specifics on how events should be triggered. Many existing algorithms [4][5][6][7][8][9] trigger events based on static thresholds applied to performance metrics such as Received Signal Strength (RSS).

For vehicular systems such approaches are limited as they do not consider how the constrained nature of movement can be used to influence predictive link triggering. We propose to capture this performance predictability through a **Fixed Route Adapted Media-streaming Enhanced handover algorithm (FRAME)**. Such an approach could be used to capture the historic experience of commuters on a route in order to optimise collective performance. Unlike other location based approaches [7][8][10][11], FRAME does not focus on the optimization of a specific handover decision. Rather, FRAME determines the optimal collective handover criteria for all Access Points (AP) on a route. Such an approach has the ability to limit the effect of spurious handovers as outlined in [12][13][14].

FRAME utilizes a directed feed forward neural network to enable MIH link triggering for multimedia streaming applications. FRAME is evaluated against the standard MIH approach [4] using performance metrics from a commercial network installation in Dublin, Ireland. Results illustrate that FRAME has a significant performance improvement over the classic MIH approach.

In this article we use frame loss rate and PSNR for performance evaluation. FRAME however, provides a pluggable extensible interface which is adaptable to emerging media stream analysis metrics and device characteristic improvements. Other media stream quality metrics such as Mean Opinion Score (MOS) or Structural Similarity (SSIM) can be easily integrated by the framework. In this work we assume the end user device has sufficient heterogeneous networking capability, battery life, available memory, and processor speed. For performance limited devices, the extensible FRAME interface allows device specific performance metrics such as those described in [15], to be utilized in the handover decision.

This paper is organised as follows: an overview of relevant mobility protocols is presented in Section II. Artificial Neural Network (ANN) concepts are introduced in Section III. Section IV describes the structure of the FRAME algorithm. The characteristics of the commercial network installation are described in Section V. Simulations and results are presented in Section VI. Related work is discussed in Section VII. Finally conclusions are discussed in Section VIII. Additional background data for this article is available from [16].

Table I: MIH Event Service – Event List

Id	Event Name	Description
1	Link Up	L2 connection is established and link is available for use
2	Link Down	L2 connection is broken and link is not available for use
3	Link Going Down	Link conditions are degrading & connection loss is imminent
4	Link Detected	New link has been detected
5	Link Parameters Change	Link parameters have crossed specified threshold
6	Link Event Rollback	Previous link event needs to be rolled back
7	Link SDU Transmit Status	Indicate transmission status of all PDU segments

II. MOBILITY PROTOCOLS

A. Media Independent Handover

MIH is a recent IEEE standard from the 802.21 working group, which supports seamless handover between homogenous and heterogeneous networks. MIH does not implement network handover; rather it provides a framework for handover between a range of network technologies including 3G, HSDPA, Bluetooth, WiFi and WiMax. MIH introduces the MIH network handover Function (MIHF) which consists of three elements: event service, command service and information service.

Media Independent Event Service

The MIH Event Service (MIES) is responsible for communicating network critical events to upper layer mobility protocols. These events are used by the upper layers to determine optimal handover time. Table I details some of the MIES events. The Link_Going_Down (LGD) event provides upper layers with a predictive indication of network degradation. FRAME focuses on the optimization of candidate path selection. As part of its optimal path selection FRAME determines, using historic performance characteristics, when path migration should occur replicating the functionality of the LGD mechanism. Many existing MIH implementations utilize a static performance threshold P_{thres} to generate the MIH LGD event. In such scenarios the relationship between the time that P_{thres} (actual or projected) is exceeded, T_{deg} , and the time at which path handover is initiated, T_{h-init} , can be expressed as follows:

$$T_{h-init} = \alpha_{lgd}(T_{deg}) \quad (1)$$

α_{lgd} is an anticipation factor applied to T_{deg} to adjust the aggressiveness of LGD event triggering. Fig 1 illustrates how such a relationship is dependent on the rate of wireless degradation.

Wireless Performance

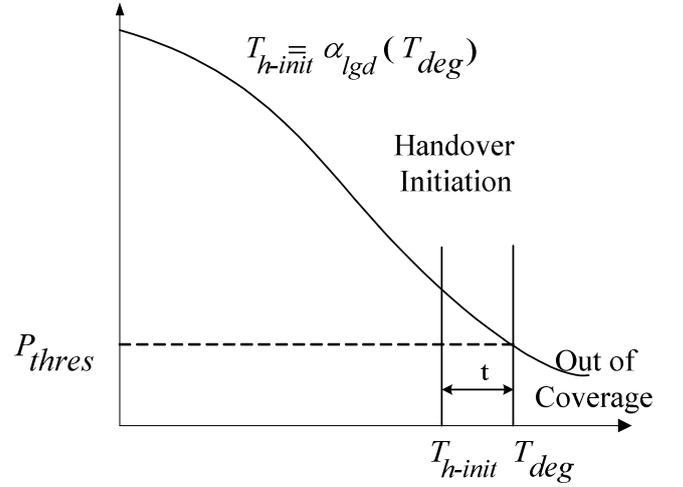


Fig. 1 Threshold-based MIH Triggering for Wireless Links

Many implementations use RSS as the indicator of network performance. If the current RSS crosses P_{thres} the LGD event is generated. The NIST MIH implementation in NS2 [4] utilizes the actual power level of packet transmission $RXThresh$ (P_{thres}) and Pr_limit (α_{lgd}) to control event triggering.

Other studies [4][5][6][7][8][9] utilize a predictive indication of RSS with P_{thres} . These approaches however, do not provide a mechanism by which the handover algorithm can tune performance thresholds for changing network conditions.

Media Independent Command Service

The MIH Command Service (MICS) enables higher layers to control the physical, data link and logical link layers. The higher layers control the reconfiguration or selection of an appropriate link through a set of handover commands. The commands carry the upper layer decisions to the lower layers. For example the MICS may be used to request a mobile node to switch between links. Commands defined include MIH_Switch and MIH_Get_Status. As FRAME assumes that SCTP [17] is the mobility protocol in use, the MIH_Switch command results in a call to SCTP's Set_Primary command. However, similar translations would be implemented for Mobile IP.

Media Independent Information Service

The Media Independent Information Service (MIIS) provides a framework by which an MIHF, residing in the mobile node or in the network, discovers and obtains network information within a geographical area to facilitate network selection and handovers. The objective is to acquire a global view of all the heterogeneous networks using metrics such as cost, QoS metrics and security. These metrics are used by a policy engine to facilitate effective handover decisions.

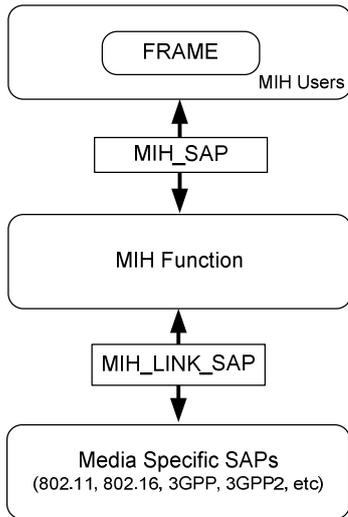


Fig. 2 FRAME/MIH Interaction

B. FRAME/MIH Interaction

Within the MIH architecture FRAME is considered an *MIH_User* which interacts with the MIHF through the MIH Service Access Point (MIH_SAP). The MIHF in turn interacts with the specific MAC through a MIH_Link_SAP. Fig 2 illustrates this interaction.

Within the MIHF, FRAME utilises the services of the MIES and MICS. Using the MIES, FRAME requests the MIHF to routinely generate a *MIH_Link_Parameters_Report.indication* service primitive which specifies the performance characteristics of specific links. The MIHF requests the MAC to generate this report using the *Link_Config_Thresholds.request* service primitive. When FRAME determines that a handover is required it requests the MICS to implement the handover using the *MIH_MN_HO_Candidate_Query.request* primitive with the required destinations ranked by preference in the *CandidateLinkList* attribute.

In FRAME, in order to maintain accurate SCTP path configurations, MIH utilises a number of MIES service primitives including; *MIH_Link_Up*, *MIH_Link_Down* and *Link_Handover_Imminent* (Layer 2 handover). The functionality of the *MIH_Link_Going_Down* service primitive is superseded by FRAME, as the proactive detection of link failure is undertaken following the reception of the *MIH_Link_Parameters_Report*.

C. Stream Control Transmission Protocol

SCTP [17] is a transport protocol originally designed to transmit signaling data across IP networks. Though it was originally designed to provide path redundancy for inter exchange signaling, SCTP has evolved to become a general purpose mobility protocol. A key feature which differentiates SCTP from TCP is multi-homing.

A multi-homed device has more than one IP address, thereby enabling a peer to contact it over a number of different physical paths within one end to end association. At start-up

SCTP selects one IP address of a multi-homed peer as the primary path for data transmission. If there are no data chunks being sent SCTP uses heartbeat packets to determine the reachability of a peer IP address. If the number of consecutive transmission timeouts exceeds the *Path.Max.Retrans* (PMR) parameter, the address is marked inactive. When a primary path becomes inactive, handover to an alternate path occurs. A number of studies have highlighted deficiencies in the SCTP switch management for mobility [18][19]. As a layer compliant protocol, SCTP is limited to using end to end metrics to gauge the performance of underlying network connections. Such an approach is performance limited for wireless networking environments. In our approach we disable the switch management features of SCTP while we still utilize its multi-homed switch implementation. By using FRAME to trigger MIH events, we avoid the performance deficiencies of a layer compliant end to end protocol.

For an MIH supported session handover time, T_{HO} , is given by:

$$T_{HO} = T_{LGD} + T_{Discovery} + T_{Selection} + T_{L3HO} + T_{L2HO} + T_{StreamAdapt} \quad (2)$$

where T_{LGD} is the time taken to generate the *Link_Going_Down* event, $T_{Discovery}$ is the time taken to identify candidate networks, $T_{Selection}$ is the time taken to select a suitable path from the set of candidates, T_{L3HO} and T_{L2HO} are the time taken to implement Layer 3 and Layer 2 handover and $T_{StreamAdapt}$ the time taken to adapt the media stream to the changed network conditions.

Many existing MIH implementations utilise Mobile IP (MIP) for switch implementation. MIP however utilises a “break before make” approach to network handover. By using the multi-homing features of SCTP it is possible to implement a “make before break” approach. If we assume the existence of a preconfigured alternate path, we can disregard the network discovery and selection components as well as the Layer 3 handover component. Using our approach, T_{HO} , is therefore reduced to:

$$T_{HO} = T_{LGD} + T_{L3HO} + T_{StreamAdapt} \quad (3)$$

III. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Network (ANN) approaches belong generally to two classes: *supervised* and *unsupervised*. Unsupervised approaches are motivated by the requirement to be autonomous self-organizing structures. Such an approach can be generally considered as the clustering of input data in order to extract useful information. The most popular unsupervised learning approaches are Adaptive Resonance Theory (ART) [20] and Kohonen networks [21]. ART networks are an evolving branch of neural networks which propose to simulate parts of the brain based on physiological models. Kohonen networks use topographic mapping in order to map high dimensional to low dimensional data.

Supervised neural networks are generally motivated by the requirements of a specific task. As the problem domain is well defined supervised approaches tend to optimize specific performance criteria. In our approach we propose to optimize the selection of candidate access points based on weightings

applied to three input parameters: RSS, link loss rate and delay. In each of our evaluations we have a clear view of optimal performance. For frame loss oriented learning the optimal performance is a 0% loss rate. For PSNR oriented learning the optimal performance is gauged against the PSNR recorded in the presence of 0% loss. The characteristics of our problem domain make it suitable for a supervised rather than an unsupervised learning. In the following sections we describe the most commonly used supervised neural network types; feed forward networks, nearest neighbor classification and radial basis functions.

A. Feed-Forward Neural Networks

The first work on ANN was presented by Mc Cullock and Pitts in 1943 [22]. They recognized that combining many simple processing units together could lead to an overall increase in computational power. The McCulloch and Pitts network had a fixed set of weights and it was Hebb [23] who developed the first learning rule. His premise was that if two neurons were active at the same time then the strength between them should be increased. Hebbian learning involves weights between learning nodes being adjusted so that each weight better represents the relationship between the nodes.

The following formula describes Hebbian learning:

$$w_{ij} = \frac{1}{p} \sum_{k=1}^p x_i^k x_j^k \quad (4)$$

w_{ij} is the weight of the connection from neuron j to neuron i , p is the number of training patterns, and x_i^k is the k^{th} input for neuron i . Classification of inputs was introduced by the perceptron model in [24]. The perceptron is a type of artificial neural network invented in 1957 by Frank Rosenblatt. As a linear classifier, it is the simplest kind of feed forward neural network. The perceptron maps its input x (a real-valued vector) to an output value $f(x)$ (a single binary value). The operation of the perceptron can be described as follows:

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

where w is a vector of real-valued weights, $w \cdot x$ is the weighted sum of inputs, and b is the 'bias', a constant term that does not depend on any input value. The value of $f(x)$ (0 or 1) is used to classify x as either a positive or a negative instance, in the case of a binary classification problem. If b is negative, then the weighted combination of inputs must produce a positive value greater than $|b|$ in order to push the classifier neuron over the 0 threshold. The bias alters the position of the decision boundary. The introduction of back propagation enabled the training of synaptic weights based on a desired output. Back propagation involved two steps: propagation and weight adjustment. Propagation involved the presentation of inputs to the ANN in order to generate output and the comparison of actual and desired output in order to generate a delta value. The weight adjustment stage involves the multiplication of each synaptic weight by a ratio of the delta value. The ratio determines the rate of learning. If the rate of learning is too small optimization can be centered on local maxima. If the learning rate is too large the ANN may never reach a trained optimal value.

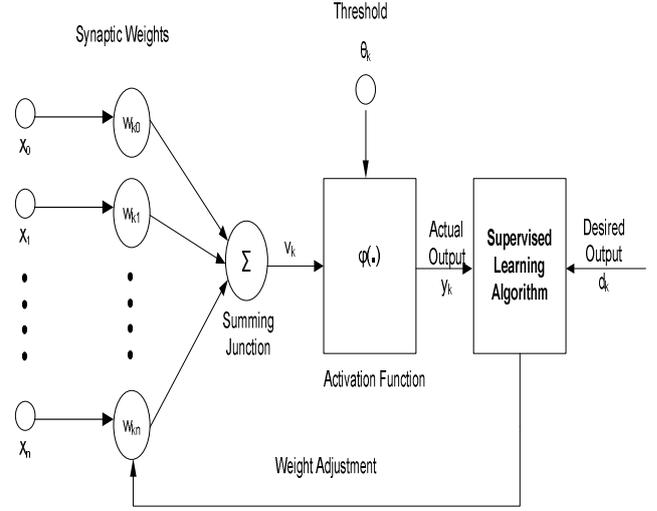


Fig. 3 A Supervised Learning Neural Network

Fig 3 illustrates a supervised learning ANN. Values $x_0, x_1, x_2, \dots, x_n$ are provided as input to the neuron. The neuron has two modes of operation: *training* or *trained*. In trained mode, the neuron applies synaptic weights $w_{k0}, w_{k1}, \dots, w_{kn}$ which enhance or degrade the input values. These weighted values are summed and an activation function $\phi(\cdot)$ is applied. $\phi(\cdot)$ determines whether the neuron should “fire”, producing an output y_k which classifies the input pattern. In supervised learning, the ANN will have an offline training phase in which neural outputs are compared against a training set. Alterations are made to the synaptic weights to limit the error in classification between the output y_k and the training set d_k . When the ANN correctly classifies the input pattern, the ANN operates in trained mode.

There are a number of common activation functions used by feed forward neural networks. A step function is a function like that used by the original Perceptron. The output is a certain value, e.g. 1, if the input sum is above a certain threshold and 0 if the input sum is below a certain threshold. These kinds of step activation functions are useful for binary classification schemes, when an input pattern has to be classified into one of two groups.

Other types of activation function include log-sigmoid and Gaussian. Fig 4 illustrates a sigmoid activation function which has the property of being similar to the step function, but with the addition of a region of uncertainty.

For FRAME we utilize a variant of the sigmoid activation function to indicate (1) if path handover should occur to a specific path and (2) the degree of certainty with which we suggest handover should occur. At routine intervals FRAME selects the network path with the highest sigmoidal activation as the primary path.

B. K-Nearest Neighbor Classification

K-nearest neighbor is a method of classifying input patterns based on their proximity to existing patterns in a training set.

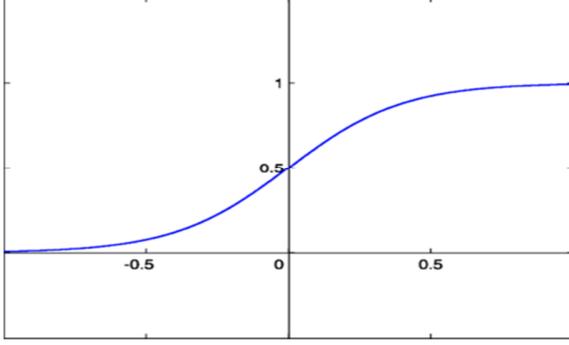


Fig. 4 Sigmoid Activation Function

The technique involves computing the distance of an input vector I_i from a set of stored training samples w_{ij} . The input pattern is assigned the classification most common amongst its k -neighbors. In its simplest form, $k=1$, the input pattern is matched against the single pattern it most closely resembles.

An approach sometimes utilized is to weigh the contributions, so that nearer neighbors contribute more to the average than more distant ones. The generalization of linear interpolation gives each neighbor a weight of $1/d$, where d is the distance to the neighbor. The selected neighbors are taken from a set of correctly classified objects.

Usually Euclidean distance is used as the distance metric. In cases such as text classification, another metric such as Hamming distance can be used. Often, the classification accuracy of k -NN can be improved significantly if the distance metric is learned with specialized algorithms such as e.g. Large Margin Nearest Neighbor or Neighbourhood components analysis.

C. Radial Basis Functions

Radial basis functions [25] emerged as a branch of ANN in the late 1980's. RBF networks typically have three layers: an input layer, a hidden layer with a non-linear RBF activation function and a linear output layer. A characteristic feature of an RBF is that their response decreases or increases with distance from a central point. Fig 5 illustrates the structure of a radial basis function ANN. The bell shaped activation functions in the hidden nodes indicate that each represents a radial basis function that is centered on a vector in the feature space. The values equidistant from the center in all directions have the same values. A critical feature which differentiates an RBF from a Multi-Layer Perceptron (MLP) is that there are no weights on the lines from the input nodes to the hidden nodes. The input vector is fed to each m -th hidden node where it is put through that nodes radial basis function

$$y_m = \exp[-\|x - c_m\|^2 / (2\sigma^2)] \quad (6)$$

where $\|x - c_m\|^2$ is the square of the distance between the input feature vector x and the center vector c_m for that radial basis function. y_m are the outputs from the radial basis functions.

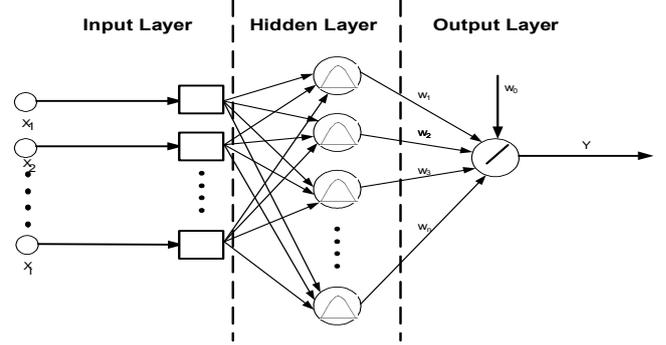


Fig. 5 Radial Basis Function ANN

IV. FRAME – FIXED ROUTE ADAPTED MEDIA STREAMING ENHANCED HANDOVER ALGORITHM

Traditionally, handover algorithms utilize static thresholds applied to metrics such as RSS [4][5]. Such approaches have evolved by proactively predicting RSS values, though still making use of static handover triggering thresholds [7][8][26].

While RSS is an important performance parameter, used alone does not provide an accurate view of the dynamic status of a link. Therefore, handover approaches have considered multiple performance parameters, including RSS, delay and loss rate. Some solutions also consider metrics related to the content delivery quality as experienced by end users [6][27].

Such approaches however are performance limited as they apply static performance thresholds, which when exceeded, trigger handover. Applying a static performance threshold makes assumptions regarding the status of a network. Previous work we have undertaken [28] illustrates that it is beneficial for the handover management algorithm to probe network performance and dynamically alter thresholds through synaptic weights. In this context, FRAME is a pluggable mechanism which can adapt to various performance metrics and alters parameter weightings based on learned behavior.

The proposed FRAME algorithm consists of two major components:

Route Identification and Management (RIM) – is responsible for the identification and management of vehicle routes. Using the vehicle geographical position, RIM identifies existing, altered or new routes.

Media Performance Directed Learning Algorithm (MPDLA) – implements the path selection intelligence within FRAME. MPDLA is a feed forward neural network which operates with a neuron dedicated to each candidate AP. Back propagation and weight adjustment are implemented each time the vehicle completes a cycle of a route.

Fig 6 outlines the pseudo code for the FRAME algorithm. FRAME dynamically configures and maintains traffic routes using GPS coordinates. Having read the GPS coordinates, FRAME determines if the current position uniquely identifies an existing route. If the position is not previously configured, a new route is created and training is initiated.

FRAME operates in either *training* or *trained* mode. As new media stream analysis metrics are likely to emerge, FRAME provides a pluggable learning mechanism, `ImplementTraining`, which allows for the utilization of alternative learning metrics. In this investigation two training

mechanisms are evaluated: frame loss rate, and PSNR-based, respectively. Initially, FRAME configures random synaptic weights in the 0.25 to 1.5 range. Typically an arbitrary activation threshold of 1 is chosen for approaches such as FRAME. Randomly allocating initial weights in the 0.25 to 1.5 range typically results in an initial configuration which is neither excessively passive nor aggressive. Subsequently, training is implemented after each route cycle. Frame loss-directed training seeks to minimize the frame loss rate. PSNR-directed training minimizes the difference between the PSNR of the video streamed on route and that of the video streamed with zero percent loss rate. FRAME is considered trained when the training process no longer updates the synaptic weights. FRAME ensures that synaptic weights remain relevant to changing network conditions by making use of a threshold, accuracyThresh.

```

Struct::Route
// Stores (a) RIM data relating GPS coordinates to a
// route (b) MPDLA parameters for the ANN
// calculations for the route
trainingmode = false //training or trained
GPS_Coord startOfRoute
GPS_Coord[] existingRoute
float[] weights // synaptic weights
float[] performanceMetric
// the list of input metrics

enum LearningType={framelosslearning,PSNRlearning}
// Which performance metric, Frame Loss/PSNR,
// is used to determine the performance of
// the network

float[] weightedMetric
// Each performance metric scaled from 0-100 *
// synaptic weight for that metric

float summedWeights
// weightedMetric[0] + weightedMetric[1]...

float activationThreshold
//if summedWeights > activationThreshold then fire

float[] historicPerf //previous throughput
float learningRate // rate of weight change
float accuracyThresh
//if throughput<accuracyThresh reinitiate training

Procedure::FRAME() // Main Procedure
GPS_Coord CurrentPos = get GPS_Position()
foreach(Route) // RIM route management
  if(CurrentPos in Route.StartOfRoute)
    // Start of a Route cycle
    historicPerf[] += perfforcurrent
    // perfforcurrent are the current performance
    // metrics relating to candidate networks
    CheckAccuracy(historicPerformance)
    if(trainingmode == true)
      ImplementTraining()
    else
      CalculateHandover()
  else if(CurrentPos in Route.ExistingRoute)
    CalculateHandover()
  else // coords will form a new route
    if(start of new route)
      // Create a new Route structure
      create Route newRoute
      newRoute.StartofRoute = CurrentPosition
      newRoute.ExistingRoute[] += CurrentPosition
      if(trainingmode == false)

```

```

    trainingmode = true
    CalculateHandover()

Procedure::CalculateHandover()
// Determine of handover is required
foreach(AP)
{
  float[] normalisedmetric
  // normalisedmetric is a performance metric scaled
  // in the range 0-100

  AP candidateAP // Potential AP for communication
  float maxActivationValue
  // The highest performing AP based on input
  // metrics and synaptic weights

  foreach(performanceMetric)
    normalisedmetric[]=NormMetric(GetPerfMetric())
    activationValue=(weights[0]*normmetric[0])+.....
    if(activationValue>threshold)
      if(activationValue>maxActivationValue)
        candidateAP=currentAP
        maxActivationValue=currentActivationValue
  implementhandover(candidateAP)

Procedure::CheckAccuracy()
// Used when FRAME is trained to ensure the trained
// synaptic weights are within acceptable bounds

  float slope = slopeofLinearRegres(historicPerf)
  if(abs(slope) > accuracyThresh)
    trainingmode = true //reinitiate training

Procedure::ImplementTraining()
// The procedure CalculteHandover() uses trained
// synaptic weights to determine if handover is
// necessary. This procedure trains those synaptic
// weights

  float errorCorrection
  // The changed which will be applied to the existing
  // weight in order to optimize performance

  int learningiteration
  // The number of iterations of learning since a last
  // random adjustment was applied. Random
  // adjustments avoid optimisation centred on local
  // maxima

  float RandomWeight
  // the size of random weight adjustment

  if(weights == null)
    randomizeweights(); // initialize random weights
  else
    float slope = slopeofLinearRegres(historicPerf)
    errorCorrection = slope*Route.learningRate
    foreach(weight in weights[])
      weight+=errorCorrection // alter weights
      if(learningiteration mode 3 ==1)
        // apply a random weight adjustment
        // after 3 cycles
        if(framelosslearning)
          RandomWeight=Rand(100-CrtFrameLoss*NormValue)
        else if(PSNRlearning)
          RandomWeight=
            Rand(MaxPSNR-CrtPSNR*NormValue)
      weight+= RandomWeight

Procedure::implementHandover(candidateAP)
// This procedure is dependant on the mobility
// protocol. It implements a call to the
// relevant mobility primitive. For SCTP this is
// the setPrimary() method
setPrimary(CandidateAP);

```

Fig. 6 Pseudo code for the FRAME algorithm

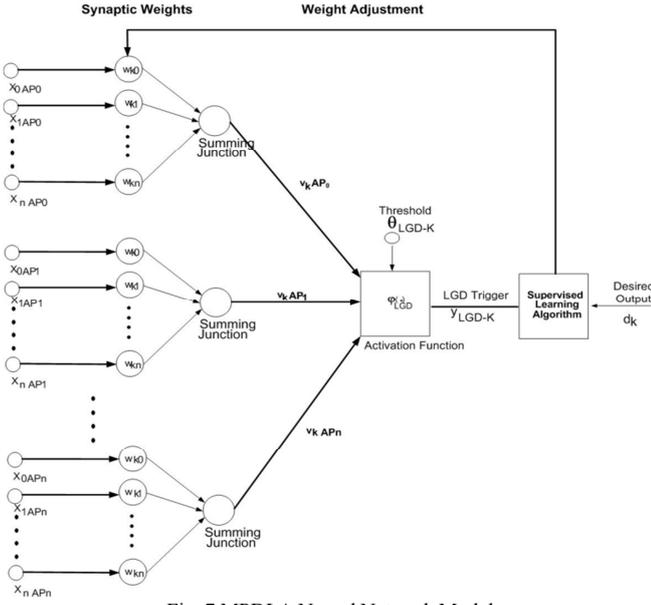


Fig. 7 MPDLA Neural Network Model

The MPDLA model consists of x_0, x_1, \dots, x_n neuron inputs corresponding to the selected performance metrics. W_{ij} is the synaptic weight applied to each performance metric j for the learning iteration i .

O_y is defined as follows:

$$V_{yAPj} = \sum_{i=0}^n X_i W_{ij} \quad (7)$$

$$O_y = \begin{cases} 1 & \text{if } \exists V_{yAPj} \geq \theta_y \\ 0 & \text{if } \forall V_{yAPj} < \theta_y \end{cases} \quad (8)$$

Fig 7 illustrates the configuration of the MPDLA model. W_{ij} are synaptic weights for each performance metric in relation to AP_j . V_{yAPj} is the sum of weighted inputs. θ_y is a user configured activation threshold. If the maximum stimulation of all neurons, $Max(V_y)$, exceeds the activation threshold θ_y , path handover occurs to the AP with $Max(V_y)$. FRAME calculates the rate of change, c , of a linear regression line for previous cycle performance as follows:

$$c = \frac{\sum(x-x')(y-y')}{\sum(x-x')^2} \quad (9)$$

c is used to gauge the effectiveness of previous synaptic weight alterations. In order to control the rate of learning, FRAME defines a user configurable learning rate constant r , where r value is between 0 and 1. The selection of an appropriate learning rate is critical for the effective operation of the algorithm. If the learning rate is too low the network learns very slowly. If the learning rate is too high weights diverge, resulting in little learning. We define the error correction, ΔW , as the product of c and r .

$$\Delta W = c * r \quad (10)$$

V. EXPERIMENTAL ANALYSIS OF FRAME IN A COMMERCIAL NETWORK

Heterogeneous networking has gained recent acceptance as the next logical step in wireless and mobile networking.

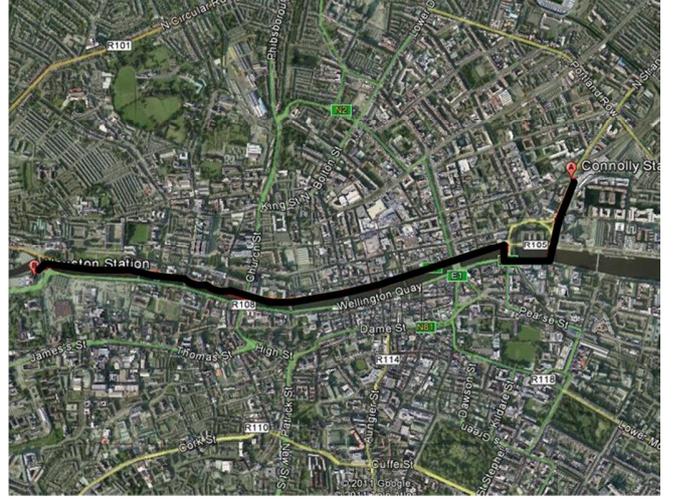


Fig. 8. Dublin City Centre Route – Connolly to Heuston Rail Stations

The International Telecommunication Union (ITU) have formalized this trend through the fourth generation (4G) set of standards [29]. Many mobile operators are embracing heterogeneous networking. Initiatives such as TeliaSonera's Homerun and British Telecom's OpenZone [30] have made heterogeneous networking a reality. In this section we analyze the performance characteristics of one such commercial deployment: Eircom's WLAN deployment in Dublin, Ireland. Eircom are the largest provider of broadband services in Republic of Ireland. Their heterogeneous network offering in Dublin City Centre consists of fixed broadband, mobile cellular (provided by Meteor Mobile) and WLAN. The handover approach proposed by FRAME is applicable to any IP network type. Previous studies we have undertaken [28] have analyzed handover between WLAN and 3G. In [28] it was assumed the 3G network was ubiquitous with relatively static performance characteristics. Handover between heterogeneous networks with dynamic characteristics such as WiMax and WLAN would require separate instances of the FRAME implementation evaluating performance metrics specific to each network type. In this work we focus on handover in a homogeneous metropolitan network.

In order to dimension the characteristics of Eircom's WLAN network we select 2 routes with varying AP concentration. Route 1 crosses Dublin City Centre from Connolly Rail station in the east to Heuston Rail station in the West. The Google Earth [31] image in Fig 8 outlines the geographical layout of this route.

Using NetStumbler [32] we record the RSS for all Eircom APs for the route outlined in Fig 8. Fig 9 illustrates the recorded RSS on a 10 minute journey by car at an average speed of approximately 18km/h from Connolly to Heuston station. As the route passes through Dublin City Centre, there is a relatively high concentration of APs. The average RSS for the duration of the test was -76.38 dBm.

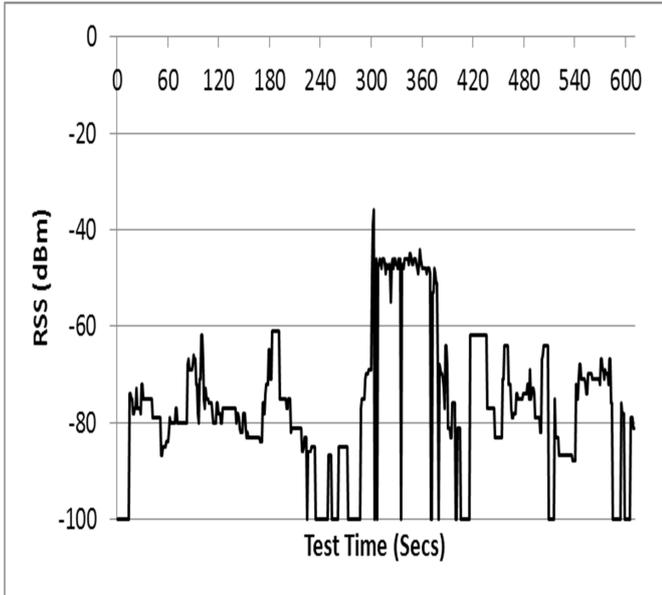


Fig. 9. Recorded RSS from Eircom APs - Dublin City Centre Route

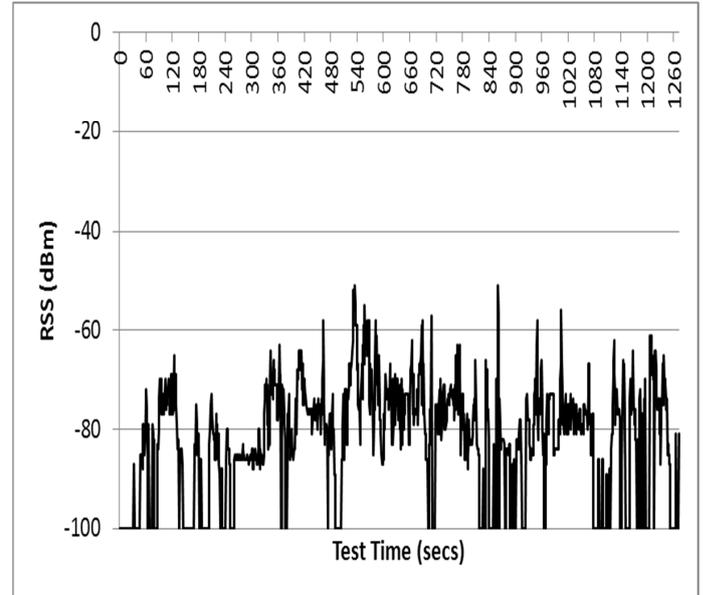


Fig. 11. Recorded RSS from Eircom APs – Dublin Airport towards Dublin City Centre



Fig. 10. Dublin Suburban Route – Dublin Airport - Dublin City Centre

The second route is through a suburban area from Dublin Airport towards Dublin City Centre. The Google Earth [31] image in Fig 10 illustrates the location of Dublin Airport in a green belt area in the north of the city.

Fig 11 illustrates the recorded RSS from Eircom APs on a 21 min journey by car at a average speed of approximately 36km/h from Dublin Airport towards Dublin City Centre.

The route from Dublin Airport to Dublin City Centre travels through a green belt area, industrial areas and suburban residential areas. The AP coverage is less dense than in the City Centre scenario. The average RSS recorded on the route was -81.94dBm in comparison to -76.38dBm recorded on the City Centre route. The experimentally recorded data was used as input to the simulation models described in the following section.

VI. SIMULATION-BASED EVALUATION OF FRAME

In this section, we evaluate the performance of FRAME using the route scenarios outlined in Section V. Our simulation uses NS2 with the MIH mobility package from NIST [32] and Evalvid [33] with the multi-homing enhancements outlined in [34]. In order to integrate the geographical location of the route into NS2, we record the GPS co-ordinates for all junctions. Using these coordinates we create two simulated versions of the routes. The first route is 3.06 Km in length and is traversed in 10 minutes. The second route is 12.07 Km in length and is traversed in 21 minutes. We simulate the streaming of the file “BigBuckBunny.cmp” in CIF format at the frame rates 17, 24 and 31FPS. This video was selected as it is of sufficiently long duration to provide content for both routes. No other characteristics of the file are relevant to our investigation. As the file has a fixed number of frames the alteration of frame rate affects the streaming duration. Increasing the frame rate reduces the streaming duration. The file streamed at 17 FPS will have a longer streaming duration than the file streamed at 24 or 31 FPS. The 17 FPS variant will therefore utilise additional elements of the network installation. Our investigation focuses on the comparison of handover strategies for the same frame rate rather than a relative comparison of the performance across different frame rates.

We recreate the RSS signatures illustrated in Fig. 9 and Fig. 11 in our simulated model. Each AP has a transmit power of 0.281838W, transmit antenna gain of 1, receive antenna gain of 1 and an antenna height of 1.5m. This provides an outdoor signal range of approximate 250m. The MIH parameters $CSThresh$ (link detection) and $RXThresh$ (link utilisation) were set to -90dBm and -85dBm, respectively. Simulation enhancements as described in [35] were included in the model. The WLAN back haul network was configured with a 100Mbps capacity and a 1ms delay. The transport layer mobility protocol SCTP was used to implement network mobility. The video file was streamed from the mobile node towards a back end content server.

Previous studies [4][5][19] have illustrated the importance of RSS, WLAN link loss rate and delay as performance metrics in a WLAN handover decision. We utilize these performance metrics as input to our FRAME algorithm. In the following sections we outline the results for both route scenarios. Each consists of (1) a brute force evaluation of optimal weight configuration based on frame loss rate, (2) an implementation of the FRAME algorithm utilizing a frame loss directed learning approach, and (3) an implementation of the FRAME algorithm utilizing a PSNR directed learning approach.

A. City Centre Route

1) Brute Force Analysis of Frame Loss Rates – City Centre Route

In this subsection we evaluate the performance of the FRAME algorithm for the City Centre scenario described in Section V in order to provide a coarse gauge of optimal weight configuration. There are 3 frame rates evaluated: 17 FPS, 24 FPS and 31 FPS. In total there are 14315 frames. Detailed results are now provided for the 24 FPS configuration. Summary results are provided for the other configurations. More detailed results can be downloaded from a results appendix available in [16]

Table II illustrates the frame loss rate for brute force tests when streaming the video in CIF format at 24 FPS. Each performance metric, Loss, delay, expressed in terms of RTT and RSS, is evaluated with their corresponding weights (w_1 , w_2 and w_3 , respectively) ranging from 0.25 to 1.5 in steps of 0.25. In general this range provided a bound of performance.

TABLE II
BRUTE FORCE FRAME LOSS NUMBER-24 FPS CITY CENTRE ROUTE

		$w_2 = 0.25$	$w_2 = 0.5$	$w_2 = 0.75$	$w_2 = 1$	$w_2 = 1.25$	$w_2 = 1.5$
$w_1 = 0.25$	$w_3 = 0.25$	14316	14316	14316	14316	14316	11700
	$w_3 = 0.5$	14316	14316	14316	14316	11696	9312
	$w_3 = 0.75$	14316	14316	14316	11696	9312	5686
	$w_3 = 1$	14316	14316	11700	9312	5686	5686
	$w_3 = 1.25$	14316	11700	9312	5670	5686	5670
	$w_3 = 1.5$	11696	9312	5670	5686	5670	5670
$w_1 = 0.5$	$w_3 = 0.25$	14316	14316	14316	12096	5340	5340
	$w_3 = 0.5$	14316	14316	12096	5340	5340	5340
	$w_3 = 0.75$	14316	12096	5306	5340	5340	3574
	$w_3 = 1$	12096	5306	5340	5340	3574	1452
	$w_3 = 1.25$	5306	5340	5340	3574	1452	1452
	$w_3 = 1.5$	5340	5340	3574	1452	1452	1452
$w_1 = 0.75$	$w_3 = 0.25$	3061	5688	5577	2797	1160	3174
	$w_3 = 0.5$	5688	5577	2797	1160	3174	1160
	$w_3 = 0.75$	5577	2797	1160	3174	1160	1452
	$w_3 = 1$	2797	1160	3174	1160	1452	1452
	$w_3 = 1.25$	1160	3174	1160	1452	1452	1452
	$w_3 = 1.5$	3174	1160	1452	1452	1452	1452
$w_1 = 1$	$w_3 = 0.25$	5149	2527	2527	2562	2767	2788
	$w_3 = 0.5$	2527	2527	2562	2767	2788	2733
	$w_3 = 0.75$	2527	2562	2767	2788	2733	3063
	$w_3 = 1$	2562	2767	2788	2733	3063	3063
	$w_3 = 1.25$	2767	2788	2733	3063	3063	3154
	$w_3 = 1.5$	2788	2733	3063	3063	3154	3061
$w_1 = 1.25$	$w_3 = 0.25$	3246	3226	3226	3226	2742	3804
	$w_3 = 0.5$	3226	3226	3226	2742	3804	3417
	$w_3 = 0.75$	3226	3226	2742	3804	3417	3440
	$w_3 = 1$	3226	2742	3804	3804	3440	3169
	$w_3 = 1.25$	2742	3804	3417	3440	3169	3361
	$w_3 = 1.5$	3804	3440	3440	3169	3361	3063
$w_1 = 1.5$	$w_3 = 0.25$	2554	2554	2527	2527	2527	2742
	$w_3 = 0.5$	2554	2527	2527	2527	2527	3405
	$w_3 = 0.75$	2527	2527	2527	2527	3405	5293
	$w_3 = 1$	2527	2527	2527	3405	5293	3485
	$w_3 = 1.25$	2527	2527	3405	5293	3485	3169
	$w_3 = 1.5$	2527	3405	5293	3485	3169	2919

Figures 12 to 17 graphically illustrate the effect of weight alterations on percentage frame loss. The initial configuration $w_1 = 0.25$ (Loss), $w_2 = 0.25$ (RTT), $w_3 = 0.25$ (RSS) resulted in a 100% frame loss rate. The 100% loss rate occurs as the output neuron does not exceed the activation threshold at any time during the cycle. Therefore no candidate AP is selected as the primary path. Increasing weights, increase the potential of exceeding the activation threshold.

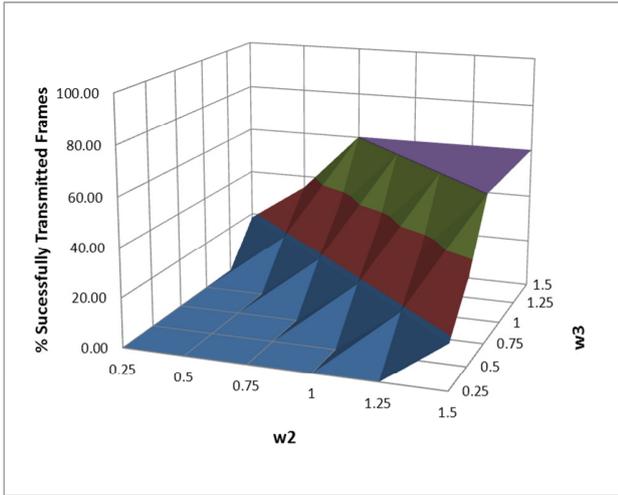


Fig. 12. Percentage of Frames Successfully Transmitted $w_1=0.25$ w_2 and w_3 vary from 0.25 to 1.5

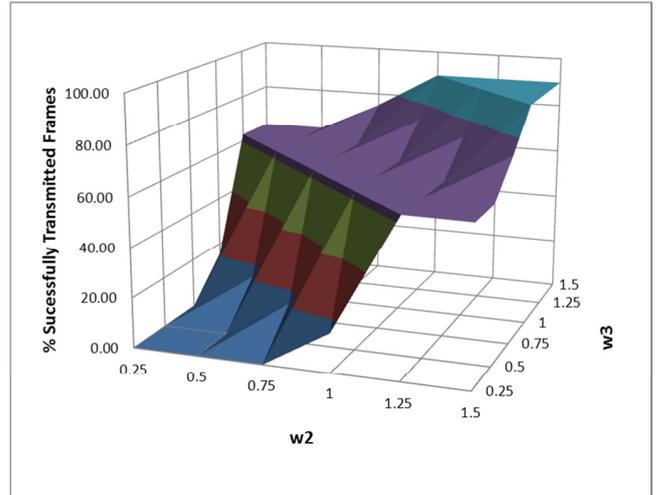


Fig. 13. Percentage of Frames Successfully Transmitted $w_1=0.5$ and w_2 and w_3 vary from 0.25 to 1.5

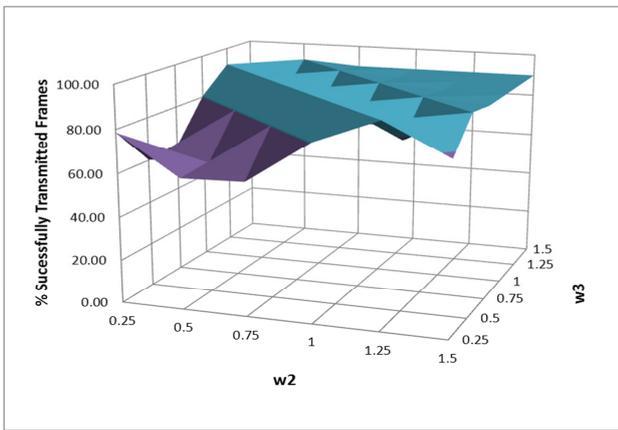


Fig. 14. Percentage of Frames Successfully Transmitted $w_1=0.75$ and w_2 and w_3 vary from 0.25 to 1.5

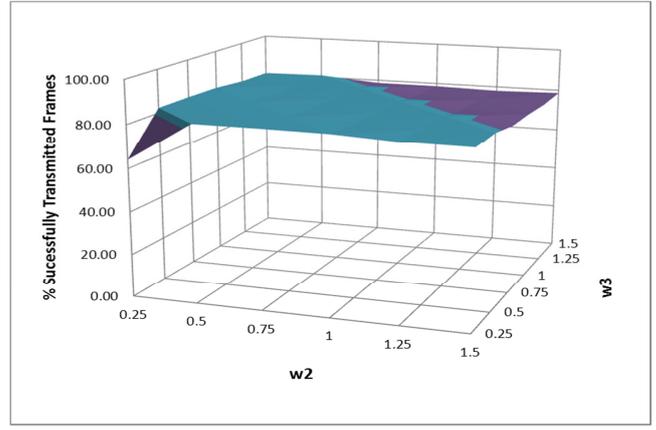


Fig. 15. Percentage of Frames Successfully Transmitted $w_1=1$ and w_2 and w_3 vary from 0.25 to 1.5

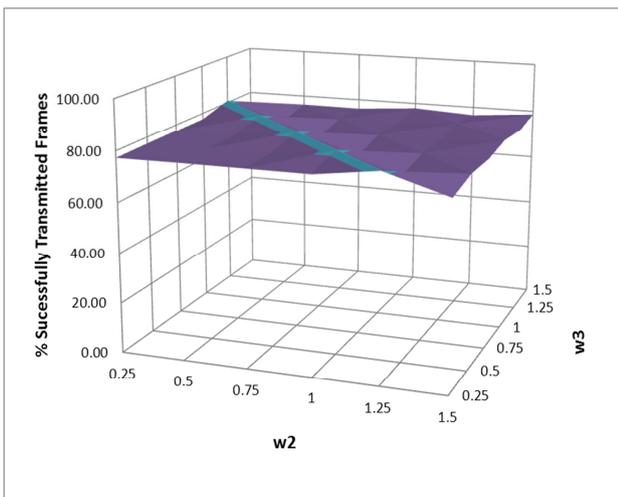


Fig. 16. Percentage of Frames Successfully Transmitted $w_1=1.25$ and w_2 and w_3 vary from 0.25 to 1.5

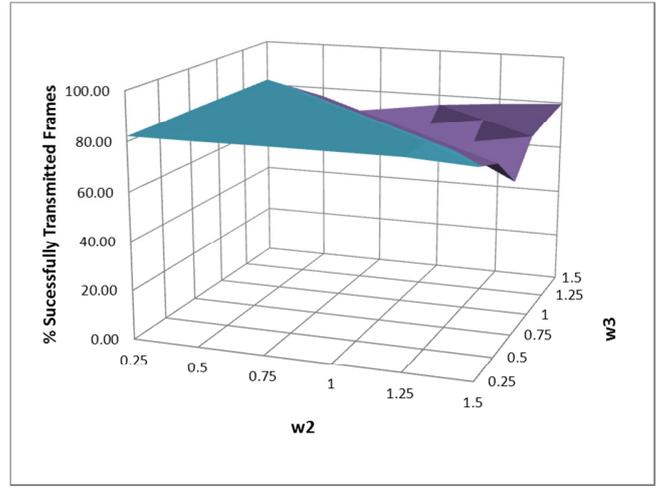


Fig. 17. Percentage of Frames Successfully Transmitted $w_1=1.5$ and w_2 and w_3 vary from 0.25 to 1.5

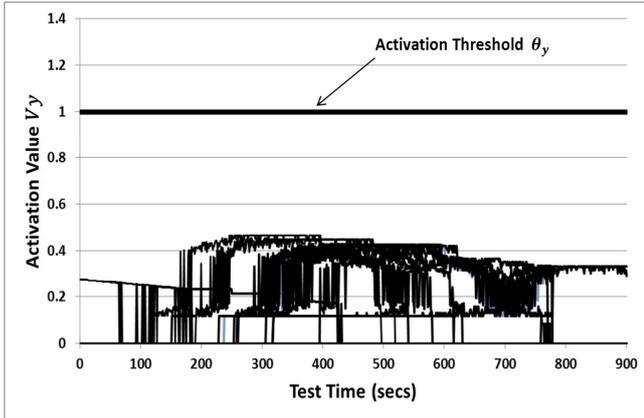


Fig. 18. Activation Value V_y $w_1=.25$ $w_2=.25$ $w_3=.25$

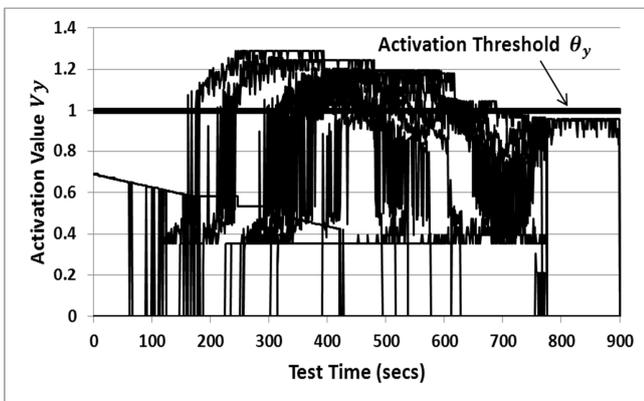


Fig. 19. Activation Value V_y $w_1=0.75$ $w_2=1$ and $w_3=0.25$

Fig 18 illustrates how when the weighting 0.25 is applied to each normalized performance metric, for each AP, the activation value V_y does not exceed the activation threshold $\theta_y = 1$ at any time.

We now consider one of the weightings which resulted in the best recorded frame loss rate of 1802 frames a percentage frame loss rate of 12.6%. Fig 19 illustrates the activation values achieved when $w_1=0.75$ (Loss), $w_2=1$ (RTT), $w_3=0.25$ (RSS). There is a period of continuous coverage in which θ_y is exceeded between 180 and 725 seconds.

For a learning algorithm such as FRAME it is important to determine the clustering of frame loss rates resulting from weight configurations. Tables A and B in the results appendix [16] detail the frame loss rates for the corresponding weight configurations for 17 and 31 FPS respectively. Using the results from Table II we define optimal frame loss in this situation as less than 2000 frames an approximate percentage frame loss rate of 14%. If there are a large number of weight configurations which result in a frame loss rate which is close to optimal this will reduce learning complexity and the number of training cycles. Inversely, if there are a large number of weight configurations which result in high frame loss rates, local maxima will reduce the effectiveness of the algorithm and increase training time.

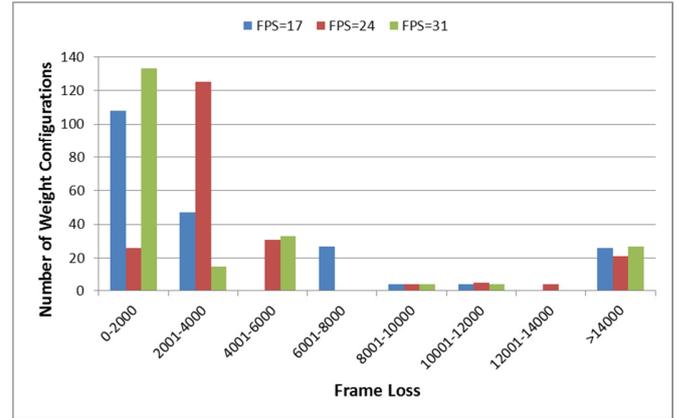


Fig. 20. Distribution of Weight Configurations

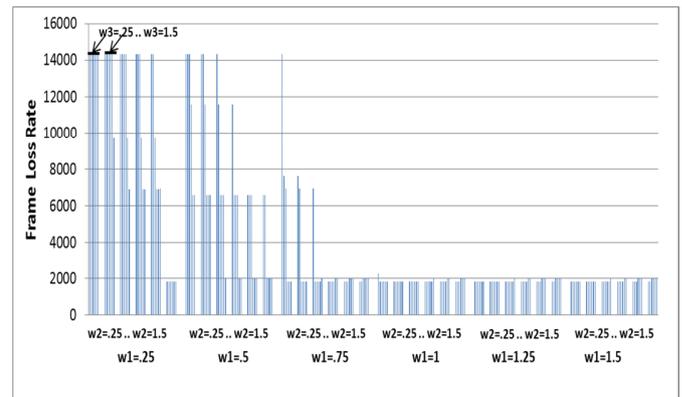


Fig. 21. 17 FPS Case - Frame Loss Rate per Weight Configuration

Fig. 20 illustrates the distribution of weight configurations and their associated frame loss. Fig. 20 illustrates that for 17 and 31 FPS there are a large number of weight configurations which result in a frame loss rate which is close to optimal (less than 2000 frames lost). This high concentration of weights resulting in optimal frame loss reduces training complexity and the number of training cycles required. For 17 FPS, 108 of the total 216 weight configurations experienced a frame loss of less than 2000 frames. For 31 FPS the number of weight configurations which experienced a frame loss of less than 2000 was 133.

For the 24 FPS case there are a large number of weight configurations which result in a frame loss of between 2001 and 4000. Such a large cluster of weight configurations centered on a suboptimal frame loss will increase training complexity. For 24 FPS, only 26 of the total 216 weight configurations experienced a frame loss of less than 2000 frames. In the following sections we illustrate how cyclical random weight adjustments can reduce the potential for local maxima.

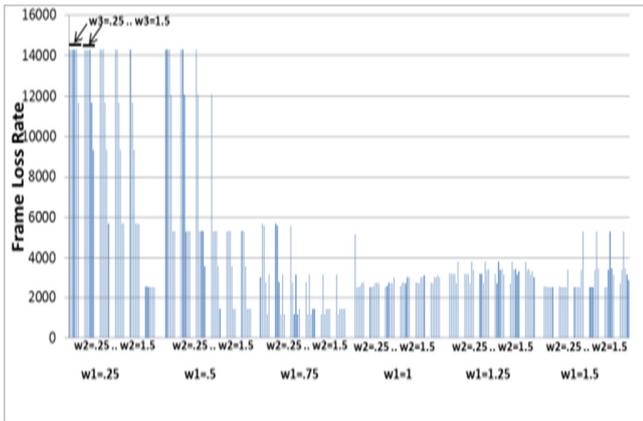


Fig. 22. 24 FPS Case - Frame Loss Rate per Weight Configuration

TABLE III
MEAN AND MODE WEIGHT CONFIGURATIONS

Mean x1 (Loss)	Mean x2 (RTT)	Mean x3 (RSS)	Mode x1 (Loss)	Mode x2 (RTT)	Mode x3 (RSS)
1.093	0.811	0.776	1.000	0.500	0.500
1.000	1.233	1.233	0.500	1.500	1.500
0.324	0.565	0.611	0.250	0.250	0.250
0.692	1.125	1.125	0.750	1.500	1.500
1.127	0.897	0.867	1.250	1.500	0.750
0.321	0.536	0.536	0.250	0.250	0.250
1.092	0.944	0.915	1.250	1.500	1.500
0.683	0.950	0.950	0.750	1.000	1.500
0.324	0.565	0.611	0.250	0.250	0.250

Fig. 21 illustrates the frame loss rates for 17 FPS with each weight w_1 , w_2 and w_3 ranging from 0.25 to 1.5 in steps of 0.25. It illustrates the dominance of the performance parameter Loss (x1). When Loss has a weighting in excess of 0.75, frame loss is typically less than 2000 frames, regardless of the weight configuration of the other performance metrics. The frame loss rate for 31 FPS follows a similar pattern.

Fig. 22 illustrates the frame loss number for 24 FPS with each weight w_1 , w_2 and w_3 ranging from 0.25 to 1.5 in steps of 0.25. Fig. 22 illustrates a greater degree of variation in frame loss rates in comparison to Fig. 21. The occurrences of optimal frame loss rates are not clustered and there it cannot be noticed a dominant performance metric. This results in a more complex training exercise.

In order to further analyze the relative importance of each performance metric, Table III presents the mean and mode weight configurations for the highest populated frame loss groups 0-2000, 2001-4000 and greater than 14000.

Table III illustrates that for the optimal frame loss rate of less than 2000, x1 (Loss) had the highest mean weighting, x2 (RTT) next highest mean weighting and x3 (RSS) the lowest mean weighting for 17 and 31 FPS. For 24 FPS the best performing weight configurations gave equally high precedence to x2 (RTT) and x3 (RSS) with weights of 1.125 while x1 (Loss) illustrating that the occurrences of optimal frame loss rates are not clustered.

We will now evaluate the FRAME algorithm in the context of these brute force test results.

2) FRAME Utilizing Frame Loss Directed Learning – City Centre Route

Table IV outlines FRAME results when employing the frame loss directed learning session with the 24 FPS video for the City Centre route. The initial weights $w_1=0.270$, $w_2=0.730$ and $w_3=0.980$ are randomly allocated. Table II illustrated that this selection of random weights is far from optimal with a low weighting for the critical parameter Loss.

As a result, there is a relatively long training session consisting of 12 cycles. The initial allocations of weights are applied to normalized performance metrics: Loss, RTT and RSS. If the activation value V_y exceeds the activation threshold $\theta_y = 1$, the neuron fires indicating that path switchover should occur.

On completion of the route cycle the frame loss rate is calculated. The first traversal of the route resulted in a frame loss number of 11700 (81.72%). If we assume that the initial loss rate for route cycle 0 was 100% we calculate the rate of change, c , of a linear regression line through both points. Using c we can determine the rate by which alterations to synaptic weights affect frame loss. For frame loss directed learning a negative c indicates that synaptic weight alterations have a beneficial effect on throughput. A positive c indicates that synaptic weight alterations have a detrimental effect on throughput. A large c (positive or negative) indicates that FRAME requires numerous training cycles. A small c indicates that the selection of weights is close to optimal. On the first cycle c has a value of -18.28 indicating that the initial weight configurations had a beneficial effect on frame loss. The error direction value, d , is used to provide an indication as to whether positive or negative c is beneficial. In this form of learning the optimal outcome is a zero frame loss rate. Therefore the FRAME algorithm encourages negative c by increasing weights. In order to relate the negative c to a positive change in weights we multiply by $d=-1$.

In order to control the rate of learning we define a user configurable learning rate constant r . The selection of an appropriate learning rate is critical for the effective operation of the algorithm. If the learning rate is too low the network learns very slowly. If the learning rate is too high weights diverge, resulting in sub-optimal learning. In this approach we use a balanced learning rate $r = 0.003$. The maximum theoretical c is -100, assuming a maximum percentage frame loss rate of 100 in learning cycle 1 and 0 percent frame loss rate in cycle 1. Therefore the maximum weight alteration achievable when $r=.3$ is $-100 \cdot -1 \cdot 0.003 = .3$. In our approach we select an activation threshold of $\theta_y = 1$. With such an activation threshold a maximum weight alteration of .3 is neither passive nor aggressive. In order to determine the alteration in weight we expand (10) to include the error direction value d :

$$\Delta w_{ij} = c * d * r \quad (11)$$

TABLE IV
FRAME EMPLOYING FRAME LOSS DIRECTED LEARNING 24 FPS CITY CENTRE ROUTE

w1 (Loss)	w2 (RTT)	w3 (RSS)	Threshold	Iteration	Frame Loss	% Frame Loss	Slope	Error Correction	Learning Rate
0.270	0.730	0.980	1	1	11700	81.72	-18.28	0.055	0.003
0.325	0.785	1.035	1	2	9312	65.04	-17.48	0.052	0.003
0.380	0.840	1.090	1	3	5222	36.47	-22.62	0.068	0.003
0.183	0.836	1.357	1	4	9784	68.34	1.65	-0.005	0.003
0.178	0.831	1.352	1	5	9784	68.34	15.93	-0.048	0.003
0.173	0.826	1.347	1	6	9780	68.31	-0.01	0.000	0.003
0.340	0.836	1.187	1	7	5316	37.13	-15.60	0.047	0.003
0.387	0.883	1.234	1	8	5316	37.13	-15.59	0.047	0.003
0.433	0.929	1.280	1	9	5653	39.48	1.18	-0.004	0.003
0.672	1.210	1.424	1	10	1452	10.14	-13.49	0.040	0.003
0.712	1.250	1.464	1	11	1452	10.14	-14.67	0.044	0.003
0.753	1.291	1.505	1	12	1452	10.14	0	0	0.003

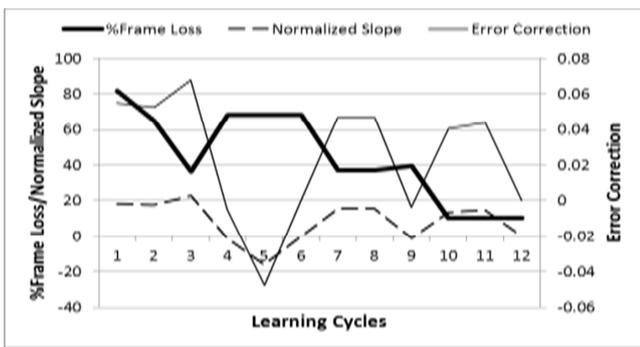


Fig. 23. 24 FPS %Frame Loss Based Learning Parameters

For the first learning cycle $c=-18.28$, $d=-1$ and $r=0.003$. we calculate $\Delta w_{ij} = -18.28 \cdot 1.003$ resulting in a weight adjustment of 0.055 for each weight. This Δw_{i1} results in a weight assignment of $w1=0.325$ $w2=0.785$ and $w3=1.035$ for the second traversal of the route. Every third cycle a random weight adjustment is applied to each weight in order to avoid the potential of learning being concentrated on local maxima. The size of this random weight adjustment is dependent on the effectiveness of the existing weight configuration. The size of the random weight adjustment is calculated as follows:

$$RandomValue * (FrameLossRate * RandomizingConstant) \quad (12)$$

The random values have a range of -100 to 100. The *RandomizingConstant* is a fixed value of 0.00008. The value 0.00008 was chosen as large frame loss rate of for example 80% will result in a random weight adjustment of the order of -0.64 to 0.64. A small frame loss rate for example 5% will result in a random weight adjustment of the order of -0.04 to 0.04. Following cycle 3 the frame loss rate was 36.47% this resulted in a random weight adjustment of the order of -0.292 to 0.292. The actual random weight adjustment were $x1 = -0.265$, $x2 = -0.072$ and $x3 = 0.199$.

For cycle 4 the weights configurations for $w1 = 0.178$ consisted of the weight for the previous learning cycle 0.380 plus the error correction 0.068 plus the random weight adjustment -0.265. The weights for $w2$ and $w3$ for cycle 4 were calculated in a similar manner. The next random weight adjustment is applied at cycle 7.

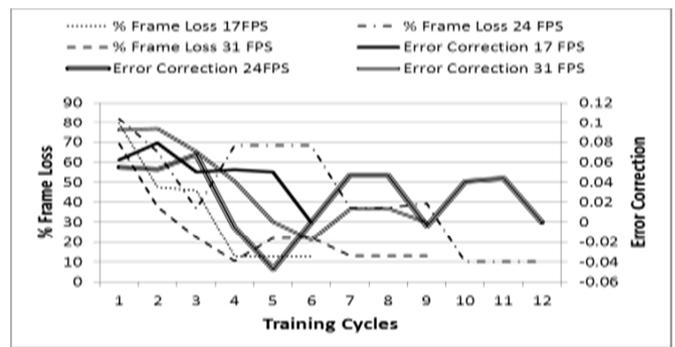


Fig. 24. % Frame Loss/Error Correction per Frame Rate

The frame loss rate for cycle 6 was 68.31%. This resulted in a random weight adjustment of the order of -0.546 to 0.546. The actual weight adjustment applied to $w1$, $w2$ and $w3$ were 0.167, 0.01 and -0.16. Fig 23 illustrates how alterations to the synaptic weights affected the frame loss rate.

Fig 23 illustrates that the random weight adjustment applied during learning cycle 4 had a detrimental effect on frame loss rate.

The frame loss number for cycle 3 was 5222 while the error correction proposed positive alterations to the weights of 0.068 the random weight of adjustment $x1 = -0.265$, $x2 = -0.072$ and $x3 = 0.199$ and final weight configuration of $x1 = 0.183$, $x2 = 0.836$ and $x3 = 1.357$ resulted in a frame loss rate of 9784 for cycle 4. This illustrates (1) the importance of link loss rate as a performance metric in handover decisions (2) the effect on overall performance of an excessively low weighting for any particular metric. Fig 23 illustrates that in the first 3 learning cycles the frame loss rate decreases as the weights of each performance metric increase. For the fourth cycle the influential parameter Loss receives a detrimental negative random weighting.

Table IV (and also Tables C and D available in [16]) illustrates the values of the training parameters for the video streamed on the City Centre route at 24 FPS (17 and 31 FPS, respectively). The 17 FPS configuration required the least number of training cycles, 6 while the 31 and 24 FPS configurations required 9 and 12 training cycles, respectively.

TABLE VI
FRAME UTILIZING PSNR DIRECTED LEARNING 24 FPS CITY CENTRE ROUTE

w1 (Loss)	w2 (RTT)	w3 (RSS)	Threshold	Iteration	PSNR	Slope	Error Correction	Learning Rate
0.53	0.77	0.33		1	15.01	15.01	0.1501	0.01
0.680	0.920	0.480	1	2	31.57	15.785	0.15785	0.01
0.837	1.077	0.637	1	3	33.78	9.385	0.09385	0.01
1.050	1.065	0.790	1	4	33.78	1.105	0.01105	0.01
1.062	1.076	0.801	1	5	33.78	0	0	0.01

TABLE V
COMPARISON OF FRAME UTILIZING FRAME LOSS DIRECTED LEARNING AND
STATIC RSS THRESHOLD LGD TRIGGERING

FPS	LGD RSS Threshold (- 85 dBm) Frame Loss	LGD RSS Threshold (-80 dBm) Frame Loss	Best Frame Loss (Brute Force Tests)	FRAME (Frame Loss Directed Learning) Frame Loss
17	8886	8846	1802	1802
24	9224	8921	1160	1452
31	9768	9269	1496	1872

The increased number of training cycles required in the 24 FPS case reflects the complexity of the training exercise as outlined in Fig. 22 together with the detrimental selection of random weight adjustment as outlined in Fig. 23. Fig. 24 compares the percentage frame loss and error correction for each of the frame rates. It illustrates a less complex learning exercise for the 17 and 31 FPS configurations in comparison to the 24 FPS situation.

Table V compares the performance of FRAME using frame loss directed learning with traditional MIH algorithms which trigger the LGD event at -85dBm and -80dBm respectively. It also compares the performance of the algorithm against the brute force test results outlined in Tables II (and Tables A and B from [16]).

Table V illustrates that the FRAME algorithm utilizing frame loss directed learning has a significant performance improvement over RSS threshold based LGD event triggering algorithms. Table A in [16] also illustrates that for 17 FPS, FRAME selected a weighting which was equal to the best frame loss identified by the brute force results. For 24 and 31 FPS respectively the algorithm selected a weighting which was 79.88% and 79.91% as effective as the best selection identified by the brute force tests.

3) FRAME Utilizing PSNR Directed Learning - City Centre Route

Table VI outlines a FRAME learning session which utilizes PSNR directed learning when streaming the media file at 24 FPS for the City Centre route. In this approach we use the PSNR of the streamed video as the basis of our learning.

As in frame loss directed learning the initial weights $w_1 = 0.530$, $w_2 = 0.770$ and $w_3 = 0.330$ are randomly allocated. Rather than taking the theoretical maximum PSNR of 100 as our optimal value, we use the maximum achievable PSNR calculated for the media file streamed with no link loss. The maximum achievable PSNR in this scenario is 36.09. The initial weight configuration $w_1 = 0.530$, $w_2 = 0.770$ and $w_3 = 0.33$ results in a PSNR value of 15.01. Assuming that the 0th

iteration had resulted in a PSNR of 0 this configuration of weights would result in a slope of 15.01. The error correction is calculated as the product of the slope and learning rate. A learning rate of 0.01 provides a theoretical error correction scale of $100 * 0.01 = 1$. In practical terms the maximum learning scale is bounded by the maximum achievable PSNR, in this situation $36.09 * 0.01 = 0.3609$. Following the first iteration a positive weight alteration of 0.1501 is applied to each of the weights. The weights for the 2nd iteration are $w_1 = 0.680$, $w_2 = 0.920$ and $w_3 = 0.48$. The weight alterations for the second and third iterations are calculated in a similar manner.

Random weight adjustments are applied following the completion of every third cycle. The size of the random weight adjustment is based on the performance of previous weight calculations and is calculated as follows:

$$\left(\frac{MAXAchievablePSNR - ActualPSNR}{MAXAchievablePSNR} \right) * RandomFactor * ScaledRandomValue \quad (13)$$

The actual PSNR is taken as a percentage of the maximum PSNR. This value is multiplied by a RandomFactor of 0.02. The value 0.02 was chosen as the random value generated is of the order -100 to 100. If a weight configuration results in a PSNR of 28.87, representing 80% of *MaxAchievablePSNR*, it will result in a random weight adjustment of the order of - 0.4 to 0.4. If a weight configuration results in a PSNR of 3.6 representing 10% of *MaxAchievablePSNR* it will result in a weight adjustment in the range of -1.8 to 1.8.

Following learning cycle 3 the PSNR is 33.78 representing 93.6% of the achievable PSNR. This PSNR value resulted in a random alteration in weight of the order of - 0.128 to 0.128. The weight configuration for iteration 4, $w_1 = 1.051$, consisted of the weight for the previous learning cycle 0.838 plus the error correction 0.093 plus a random weight adjustment of 0.119. The corresponding random weight adjustments for w_2 and w_3 were - 0.107 and 0.059.

Fig. 25 illustrates the achieved PSNR per frame for the configurations with no link loss, RSS threshold LGD triggering at -85 dBm and the best performing weight configuration $w_1 = 1.062$, $w_2 = 1.076$ and $w_3 = 0.801$. It illustrates that a deviation in PSNR occurred between the weight configuration $w_1 = 1.062$, $w_2 = 1.076$ and $w_3 = 0.801$ and the maximum achievable PSNR between frame 1 and frame 1163. Fig 25 also illustrates that the tradition LGD triggering approach based on a threshold RSS of -85dBm has significant performance degradation between frame 1 and 3287 and again between frame 8280 and 14317.

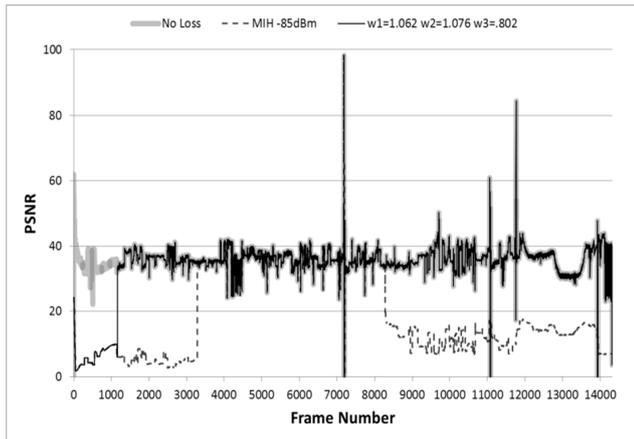


Fig. 25. Frame Loss/Error Correction per Frame Rate

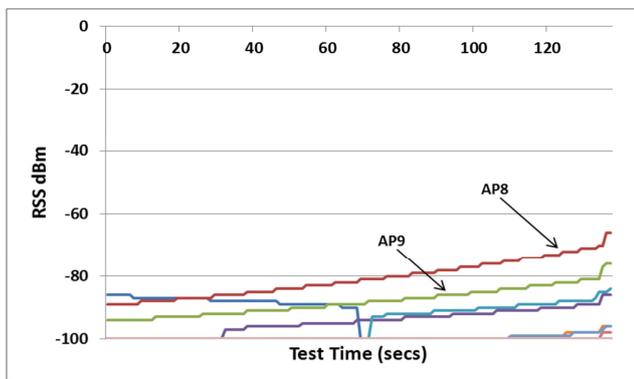


Fig. 26. Candidate AP RSS

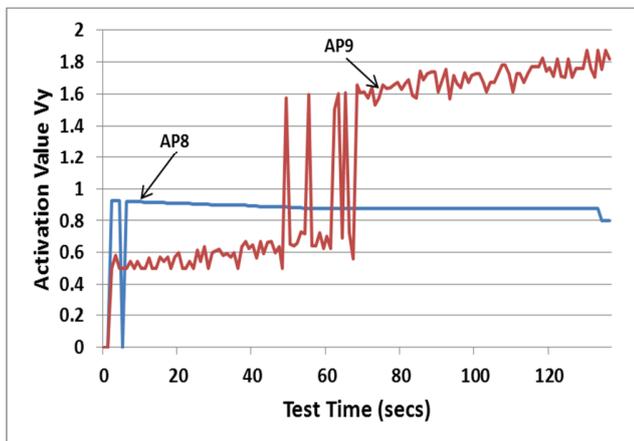


Fig. 27. Activation Values of Candidate AP8 and AP9

At a frame rate of 24 FPS the performance degradation for the FRAME configuration is between $t=1$ second to $t=48.4$ seconds. For the RSS based LGD triggering approach the performance degradation occurs between $t=1$ second and $t=136.9$ seconds and again between $t=345$ seconds and $t=596.5$ seconds. In order to analyze the performance degradation we consider the characteristics of the candidate networks during this period in more detail. Fig. 26 illustrates the RSS for all available APs for the period 1-137 seconds.

TABLE VII
COMPARISON OF FRAME UTILIZING PSNR DIRECTED LEARNING AND TRADITIONAL RSS THRESHOLD LGD TRIGGERING

FPS	Max PSNR (No Link Loss Rate)	MIH - 85dBm PSNR	MIH - 80dBm PSNR	FRAME Learning PSNR
17	36.09	19.52	20.11	33.78
24	36.09	18.74	19.05	33.78
31	36.09	16.8	17.73	32.22

TABLE VIII
COMPARISON OF FRAME WITH FRAME LOSS AND FRAME WITH PSNR DIRECTED LEARNING APPROACHES

FPS	Learning Type	Final Optimal Weights			Frame Loss Achieved	PSNR Achieved
		w1 (Loss)	w2 (RTT)	w3 (RSS)		
17	Frame Loss	0.879	1.136	0.807	1802	32.64
	PSNR	1.0516	0.776	1.201	1802	33.78
24	Frame Loss	0.753	1.291	1.505	1452	32.95
	PSNR	1.062	1.076	0.801	1160	33.78
31	Frame Loss	0.845	1.150	1.160	1872	32.22
	PSNR	0.781	1.102	1.231	1872	32.22

AP₈ has higher RSS and is therefore selected by the traditional RSS threshold approach as the AP of choice. If an RSS threshold of -85dBm is applied, AP₈ is selected as the primary path at $t=45$ seconds. If an RSS threshold of -80dBm is applied the AP₈ is selected as the primary after 82 seconds. AP₉ is never considered as the primary as its RSS is less than that of AP₈ at all times during this period.

Fig. 27 illustrates the activation values for the weight configuration $w_1=1.062$, $w_2=1.076$ and $w_3 = 0.801$. It illustrates a significant divergence in performance when the parameters loss and delay are considered in conjunction with RSS.

Initially AP₈ has the highest activation value, though not sufficient to exceed the activation threshold of 1. The activation value for AP₉ momentarily exceeds the activation threshold at 46, 51 and 59 seconds before a final selection of AP₉ as primary at 61 seconds.

Fig. 27 illustrates that by dynamically dimensioning a number of performance metrics, the FRAME algorithm has significant improvements over static threshold based LGD triggering mechanisms.

Tables E and F [16] illustrate the final trained values for the media file streamed on the City Centre route at 17 and 31 FPS utilizing PSNR directed learning. Tables VI, E and F illustrate that all configurations required relatively few learning cycles 5-6. Table VII compares the performance of FRAME PSNR directed learning approaches and static threshold based LGD triggering mechanisms based on RSS.

Table VII illustrates that FRAME with PSNR directed learning has significant performance improvements over static threshold based LGD triggering mechanisms. The 17 FPS configuration had a 73% and 68% performance improvement over the LDG triggering approaches based on an RSS threshold of -85dBm and -80dBm respectively.

TABLE IX
BRUTE FORCE FRAME LOSS RATE 24 FPS SUBURBAN ROUTE

		w2 = 0.25	w2=0. 5	w2= 0.75	w2=1	w2= 1.25	w2= 1.5
w1=0.25	w3=0.25	14316	14316	9072	7827	8430	8522
	w3=0.5	14316	9072	7827	8430	8522	8522
	w3=0.75	9072	7827	8430	7478	8522	8522
	w3=1	7827	8430	7478	8522	8522	8522
	w3=1.25	8430	8522	8522	8522	8522	8522
	w3=1.5	7478	8522	8522	8522	8522	8522
w1=0.5	w3=0.25	14316	8444	7343	7451	8430	8430
	w3=0.5	8444	7343	7451	8521	8430	8430
	w3=0.75	7343	7451	8521	8430	8430	8430
	w3=1	7451	8521	8430	8430	8430	8430
	w3=1.25	8521	8430	8430	8430	8430	8430
	w3=1.5	8430	8430	8430	8430	8430	8430
w1=0.75	w3=0.25	8414	7508	8450	7343	7397	7397
	w3=0.5	7508	8450	7343	7343	7397	7451
	w3=0.75	8450	7343	7343	7397	7451	8521
	w3=1	7343	7343	7397	7451	8521	8430
	w3=1.25	7343	7397	7451	8430	8430	8430
	w3=1.5	7397	7451	8430	8430	8430	8430
w1=1	w3=0.25	8366	7508	7508	8450	8450	7343
	w3=0.5	7508	7508	8450	8450	7343	7343
	w3=0.75	7508	8450	8450	7343	7343	7397
	w3=1	8450	8450	7343	7343	7397	7451
	w3=1.25	8450	7343	7343	7397	7451	8521
	w3=1.5	7343	7343	7397	7451	8521	8521
w1=1.25	w3=0.25	6866	6942	6942	6942	6863	6863
	w3=0.5	6942	6942	6942	6863	6863	7093
	w3=0.75	6942	6942	6863	6863	7093	7073
	w3=1	6942	6863	6863	7093	7073	7073
	w3=1.25	6863	6863	7093	7073	7073	7046
	w3=1.5	6863	7093	7073	7073	7046	7046
w1=1.5	w3=0.25	6884	8027	7100	7100	8072	8072
	w3=0.5	8027	7100	7100	8072	8072	8072
	w3=0.75	7100	7100	7100	8072	8072	8072
	w3=1	7100	7100	8072	8072	8072	7103
	w3=1.25	7100	8072	8072	8072	7103	7103
	w3=1.5	8072	8072	8072	7103	7103	7103

The 24 FPS configuration had an 80% and 77% performance improvement. The 31 FPS configurations had the most significant performance improvement with a 92% and 82% performance improvement.

Table VIII details the FRAME final synaptic weights for frame loss and for PSNR directed learning. Fig. 21 illustrated that for 17 FPS, a large number of weight configurations resulted in close to optimal frame loss. While the weight configurations in both learning methods vary, the frame loss rate achieved is identical and the final PSNR in both methods is comparable. The PSNR approach has a 3.3% performance improvement over the frame loss approach. For 24 FPS, the PSNR approach had 20.1% less frame loss rate than the frame loss approach. When the final PSNR is considered however, the performance improvement was only 2.5%. Table VIII illustrates that both learning methods have equivalent performance for the 31 FPS configuration.

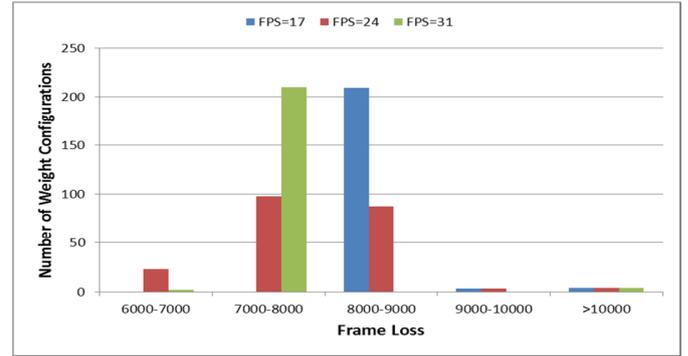


Fig. 28. Distribution of Weight Configurations

TABLE X
MEAN AND MODE WEIGHT CONFIGURATIONS

		Mean X1	Mean X2	Mean x3	Mode X1	Mode x2	Mode X3
FPS=17	6000-7000	N/A	N/A	N/A	N/A	N/A	N/A
	7000-8000	N/A	N/A	N/A	N/A	N/A	N/A
	8000-9000	0.895	0.891	0.891	1.500	1.500	1.000
FPS=24	6000-7000	1.217	0.685	0.630	1.250	0.250	0.250
	7000-8000	0.946	0.887	0.884	1.000	1.500	1.000
	8000-9000	0.756	0.949	0.966	0.500	1.250	1.500
FPS=31	6000-7000	0.875	0.875	0.250	1.500	0.250	0.250
	7000-8000	0.886	0.886	0.892	1.250	1.250	1.500
	8000-9000	0.313	0.313	0.313	0.250	0.250	0.250

The aim of the FRAME algorithm is to provide a mobile multimedia user with seamless handover. Table VIII illustrates that the more computationally complex PSNR learning method has some improved performance over the frame loss learning approach. However, the selection of an appropriate learning method is an implementation specific tradeoff between learning mechanism complexity, particularly for memory constrained devices, and the level of performance improvement achievable.

B. Suburban Route

1) Brute Force Analysis – Suburban Route

In the previous section we analyzed the performance of FRAME in a City Centre configuration with a high density of APs. In this section we evaluate the performance of FRAME in a suburban environment from Dublin Airport to Dublin City Centre, as illustrated in Fig. 10. The route from Dublin Airport to Dublin City Centre is categorized by green belt, industrial and suburban development areas. The AP coverage is therefore less dense than in the City Centre scenario.

Table IX illustrates the frame loss rate for brute force weight configurations when streaming the media file in CIF format at the default frame rate of 24 FPS. Tables G and H [16] indicate the corresponding frame loss rates for 17 and 31 FPS respectively.

Fig 28 illustrates the distribution of weight configurations and their associated frame loss for the suburban route. Fig. 20 and Fig. 28 illustrate how the density of APs in the City Centre versus the suburban route affects frame loss. In the City Centre scenario the greatest density of weight configurations result in a frame loss number of 0-2000 frames.

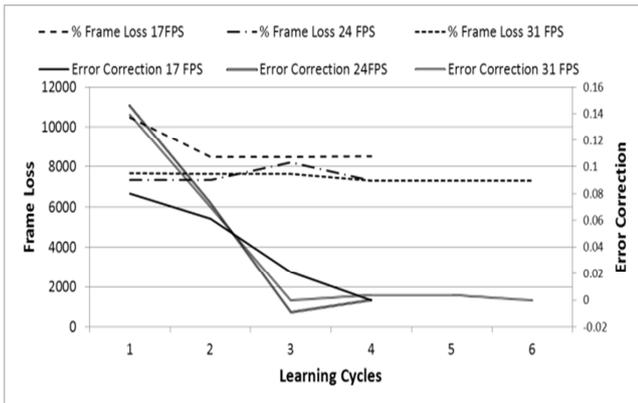


Fig. 29. Frame Loss/Error Correction 17, 24 and 31 FPS

TABLE XI
COMPARISON OF FRAME UTILIZING FRAME LOSS DIRECTED LEARNING AND TRADITIONAL RSS THRESHOLD LGD TRIGGERING

FPS	LGD RSS Threshold (-85 dBm) Frame Loss	LGD RSS Threshold (-80 dBm) Frame Loss	Optimal Frame Loss (Brute Force Tests)	FRAME (Frame Loss Directed Learning) Frame Loss
17	9487	9280	8209	8524
24	8408	8783	6863	7298
31	8281	7643	6982	7297

Fig. 28 illustrates that for the suburban route the greatest density of weight configurations result in a frame loss number of 7000-9000 frames. Table X illustrates the mean and mode for each of the frame rates for the intervals 6000-7000, 7000-8000 and 8000-9000.

Table IX illustrates that no weight configuration resulted in a frame loss rate of less than 6000 frames. In order to explain this high frame loss we analyze the RSS signature in Fig. 11.

Fig. 11 illustrates an initial high signal coverage close to the airport. As the vehicle moves from the airport, there is intermittent signal coverage for approximately 6 minutes until we approach the City Centre. For a 17 FPS video sequence the duration of the test is 842 seconds. For 24 and 31 FPS the duration of the test is 597 and 462 seconds respectively. Due to higher frame rates the 24 and 31 FPS configurations benefit more from the initial signal coverage at the airport.

2) FRAME Utilizing Frame Loss Directed Learning -Suburban Route

The reduced AP concentration on the suburban route increases the frame loss rate. Tables J, K and L illustrate the final trained values for the media file streamed on the suburban route at 17, 24 and 31 FPS. The average frame loss rate on the City Centre route was 4869 while the average frame loss number on the suburban route was 7885. FRAME uses dynamic probing to determine network conditions. In the City Centre, the high availability of candidate APs enables FRAME to proactively assess performance and provides significant performance improvement over static threshold based MIH event triggering approaches.

TABLE XII
COMPARISON OF FRAME UTILIZING FRAME LOSS DIRECTED LEARNING AND TRADITIONAL RSS THRESHOLD LGD TRIGGERING

	FPS	MIH - 80dBm Frame Loss	FRAME Frame Loss Rate	% Performance Improvement
City Centre	17	8846	1802	390.90
	24	8921	1452	514.39
	31	9269	1872	395.14
Airport - City Centre	17	9280	8524	8.87
	24	8783	8450	3.94
	31	7643	7297	4.74

In the suburban configuration the reduced AP availability reduces the path selection complexity and limits the potential performance improvement. This results in a reduced number of training cycles. The 17 and 31 FPS configurations both require 6 training sessions while the 24 FPS configuration only requires 4 training sessions.

Fig. 29 illustrates the frame loss rate and error correction for 17, 24 and 31 FPS for each learning cycle. It illustrates that the high frequency of weights with frame loss numbers in the range 7000-9000 causes the weights to converge to an optimal value quickly.

Table XI compares the performance of FRAME using frame loss directed learning in comparison to static threshold based LGD triggering mechanisms based on RSS.

It also compares the performance of the frame loss directed learning method against the best performing weight configurations outlined in Tables J, K and L.

Table XI illustrates that FRAME with frame loss directed learning had a 10%, 13% and 12% performance improvement over a static RSS event threshold of -85dBm for 17, 24 and 31 FPS respectively. It also had an 8%, 17% and 5% performance improvement over a static RSS event threshold of -80dBm for 17, 24 and 31 FPS respectively.

For 17, 24 and 31 FPS respectively, FRAME selected a weighting which was 96.3%, 94.0% and 95.6% as effective as the optimal selection identified by the brute force analysis outlined in Tables IX, G and H.

Table XI does however illustrate that while FRAME has a performance improvement it is not as significant as that experienced in the City Centre scenario. The variation in performance relates to the higher availability of candidate APs in the City Centre route and demand for a more complex mobility management decision in such a scenario.

Table XII compares the performance of FRAME with frame loss directed learning and static RSS threshold based LGD event triggering for both the City Centre and suburban scenarios. It illustrates that in the City Centre scenario where there was a high availability of candidate APs, FRAME had between 391% and 514% performance improvement over traditional LGD link triggering mechanisms. In the suburban route where AP coverage was limited the performance improvement was less notable at between 3.9% and 8.9%.

TABLE XIII
COMPARISON OF FRAME UTILIZING PSNR DIRECTED LEARNING AND
TRADITIONAL RSS THRESHOLD LGD TRIGGERING

FPS	Max PSNR (No Link Loss Rate)	MIH - 85dBm PSNR	MIH - 80dBm PSNR	FRAME Learning PSNR
17	36.09	19.75	19.33	22.03
24	36.09	20.87	19.34	22.55
31	36.09	23.05	21.25	23.01

TABLE XIV
COMPARISON OF FRAME UTILIZING FRAME LOSS AND FRAME UTILIZING
PSNR DIRECTED LEARNING APPROACHES

FPS	Learning Type	Final Optimal Weights			Frame Loss Achieved	PSNR Achieved
		w1 (Loss)	w2 (RTT)	w3 (RSS)		
17	Frame Loss	0.224	0.780	0.746	8524	20.36
	PSNR	1.875	1.032	0.541	8257	22.03
24	Frame Loss	1.262	0.704	1.034	7298	23.01
	PSNR	1.709	1.586	0.844	8071	22.55
31	Frame Loss	1.095	0.741	1.160	7297	23.01
	PSNR	1.583	0.635	1.117	7297	23.01

3) FRAME Utilizing PSNR Directed Learning - Suburban Route

Table M, N and O [16] illustrate the final trained values for the media file streamed on the suburban route at 17, 24 and 31 FPS for FRAME with PSNR directed learning.

They illustrate that all configurations required relatively few learning cycles 5-6. Table XIII compares the performance of FRAME with PSNR directed learning and static threshold based LGD triggering mechanisms for the suburban route.

Table XIII illustrates that the FRAME PSNR directed learning approaches have generally better performance than the static threshold based LGD triggering mechanisms with performance improvement ranging up to 14%.

Table XIV details the final synaptic weights for FRAME utilizing frame loss and PSNR directed learning for the suburban route. It illustrates that FRAME has equivalent if not improved performance over the more computationally intensive PSNR learning method. In an environment where end user devices have constrained characteristics, the computationally light metric frame loss provides an accurate assessment of final PSNR. For the 17 FPS configuration, the PSNR directed learning approach has an 8.2% performance improvement. For the 24 FPS configuration, the frame loss directed learning approach has a 2% performance improvement over the PSNR directed learning approach. For the 31 FPS configuration both learning methods have equivalent performance.

VII. RELATED WORK

Location aware systems are a type of context aware systems which attempt to predict the end user movement in order engineer network performance [36][37]. The confined nature of movement in vehicle based systems simplifies the predictability of end user movement. A number of studies

exploit this end user movement predictability in order to optimize Vertical Hand Over (VHO). [10] Introduces a mechanism which divides a geographical area into zones and uses previous historical behavior to determine the most likely zone for handover. In [8] the authors propose to determine the end location of the end user in order to assist handover decision. In [11] the GPS coordinates on the vehicle are traced in order to estimate potential network migrations based on the MN's motion. An adaptive approach based on predicted RSS is presented in [7]. These approaches are primarily concerned with determining network coverage. They do not consider how the dynamic characteristics of candidate networks could affect end user QoS. It may be useful to determine when a MN will enter the coverage of wireless or mobile network. However, if that network is congested it may not be an appropriate candidate for path selection. The FRAME algorithm exploits predictable movement patterns while also considering the dynamic performance characteristics of available networks.

A number of studies consider both the coverage prediction provided by location aware approaches and the dynamic characteristics of the available networks. [9] considers both the device's location information and packet arrival time. In [38] a location-based Vehicle Handover Algorithm (VHA) is proposed which combines both mobile location and network information in order to limit spurious handovers thereby improving VHO latency. All of the approaches outlined are concerned with the optimization of handover between specific access networks. The current position of the MN and its direction is typically used to estimate when the MN will leave the coverage of the current network and enter the coverage of another. The FRAME algorithm is an abstracted approach which determines the optimal collective handover criteria for all APs on a route. The handover decision implemented by FRAME maybe less effective in certain specific handover decisions. However, by analyzing the data for the route in its entirety collective performance is optimized. Such an approach has the ability to limit the effect of spurious handovers as outlined in [38].

Various studies have investigated algorithms to optimize MIH event triggering. [39] proposes an approach which can alter the LGD trigger threshold based on the estimated time of entry into the next target cell. In [40] additional primitives for MIH are proposed. A new primitive "MIH-PrefixInfo" is defined and a parameter "prefix" is added to the existing MIH primitives to reduce handover latency. Predictive handover mechanisms are proposed in [41], which use the network neighbourhood information in the MIIS to prepare for the impending handover before failure of the current network. [42] proposes an integration architecture for the MIIS, in which several networks elements collaborate in the discovery of network information. MNs collect the desired neighboring network information with a query-response mechanism. Results illustrate that the approach shortens network selection time. An enhanced Media Independent Handover Framework (eMIHF) is designed in [43]. This implementation extends MIH by allowing for efficient provisioning and activation of QoS resources during the handover preparation phase. In [44] a data rate based vertical handover triggering mechanism (DR-HTM) is introduced for MIH. When a MN discovers a

candidate WLAN, it obtains its achievable data rate using remote MIH services. Results illustrate that overall network utilization can be improved using approach. In [45] it is suggested that MIH is not aware of different user contexts, and therefore cannot provide context-aware services to improve user experience. To address this problem a context-aware module is introduced. The context aware module is responsible for generating timely MIH Link Going Down (LGD) allowing sufficient duration for session adaptation

A number of studies have integrated location based awareness with MIH event triggering. [64] proposes a Seamless Wireless internet for Fast Trains (SWiFT) algorithm. SWiFT introduces a L2 MIH trigger which uses a handover probability value calculated using current RSS and the speed of movement. [46] analyses the relationship between mobile node speed, cell coverage and MIH L2 event triggering time for a MIP based implementation. The study reiterates the importance of effective L2 link triggering in order to limit packet loss and delay. In [47] a Mobile Stream Control Transmission Protocol (MSCTP)-based handover scheme for Vehicular Networks (VNs) is introduced. The approach utilises MIH for switch management and SCTP for switch implementation. During WLAN path migration an alternate cellular path is utilised. A number of new SCTP chunk types are introduced including Address Configuration Change (ASCONF)-ChangePrim, ASCONF-RegisterIP, and ASCONF- Path-Switching chunk. Such an approach however may not be appropriate as previous work we have undertaken [48][49] illustrates the potential for receiver buffer blocking communication failure when paths have significantly differing performance characteristics.

In [50] an adaptive QoS to improve media streaming service performance is introduced targeted towards vehicular systems. Results presented indicate that handover time is actually increased using the mechanism though overall throughput is reduced. For evaluation the authors use the standard MIH approach outlined in NS2. This model utilise Mobile IP as the mobility protocol. In Section II we outlined the potential performance improvement of using a multi-homed “make before break” approach rather than the “break before make” approach provided by an exclusively MIP oriented approach. In [51] a media streaming application is built which uses MIH. The approach proposes to assure media streaming service continuity in heterogeneous networking environments. A performance evaluation of the approach is undertaken using audio streaming continuity assessment. A buffer delay of 2 seconds is recorded making the approach unsuitable for real time streaming applications. The authors do not provide detail on the mobility protocol employed. MIH is a framework for the communication of network critical events to upper layer mobility protocols. MIH does not (a) implement mobility (b) provide an implementation of how events should be triggered. In order to assess the performance of an MIH implementation it is necessary to specify the mechanisms employed for both (a) and (b).

Some approaches propose to utilize mSCTP for both switch management and switch implementation. In [52] the authors identify a potential issue with regard to the transmission of the SCTP ASCONF packet prior to catastrophic network failure. An optimized vertical handover

approach is suggested which alters the SCTP SACK chunk in and retransmits the ASCONF immediately following handover. In [53] an enhancement to mSCTP is proposed which performs primary path switching before it becomes unavailable due to its primary path drop. The improvement of the scheme comes from considering the temporal velocity of the mobile terminal with relative RTT variances. Results we presented in [19] and [49] highlight a potential performance issue with such an approach. In wireless environments RTT can increase in an exponential manner as a MN moves from the coverage area. On entering the coverage of a wireless AP the RTT may initially be very high and then decrease quickly. This behaviour relates to the number of transmissions in the MAC layer. RTT as a performance metric be effected by this behaviour. Results we have presented [19][49] indicate that the end to end nature of a transport layer compliant solution such as SCTP make it unsuitable as a switch management approach. In the above approach, the detection of path failure will be at least one RTO. In wireless environments this delay can be significantly longer. It is unlikely that a SCTP specific approach can meet the delay requirements of a media streaming application. In [54] the authors propose a handoff scheme for vehicular based systems which utilizes MIP for network discovery and address configuration. The approach utilizes SCTP for switch implementation. Switch management is based on the RSS of beacon frames. MN direction and speed is considered in the switch decision. Such an approach has similarities with location aware mobility solutions. In such an approach the performance characteristics of candidate APs are not considered. This may lead to a potentially detrimental switch decision. A similar approach is outlined in [9] which uses the RSS of available AP as the basis for MIH event triggering.

A number of studies have applied ANN mechanisms to computer communications. Approaches such as the Kohonen Self Organizing MAP (SOM) assist network management personnel in the visualization of network conditions. Other approaches are concerned with network load balancing where network handover is initiated to implement the load balancing decision. Our focus relates to ANN approaches which optimize handover for specific end points.

In [56][57][58] SOM methods are introduced to ease the operation of mobile networks by introducing advanced monitoring and automated optimisation methods. These methods are used to detect mal-functioning network elements as well as clustering network elements in order to implement parameter optimisation on network element-cluster level. In [56] and [57] the authors investigate how SOM methods can be utilised for network host intrusion detection. [56] concentrated on advanced visualisation of intrusion while [57] analysed the mechanisms by which the intrusion is detected. In [58] the application area was extended to mobile network cell performance monitoring, clustering and anomaly detection. The results presented indicated that the methods were suitable for anomaly detection, visualization and clustering. Such unsupervised methods however are unlikely to provide the level of performance granularity required to determine handover initiation time for a specific end user.

Significant ANN research focuses on load balancing for heterogeneous networks. In [59] and [60] the authors propose

a mutually connected neural network in order to optimise load balancing and QoS for the entire network. In [61] the authors propose a self-adaptive K value selection scheme for optimizing load balancing in large-scale 802.16 systems. In [62] the authors utilise a radial basis function ANN to classify the Uplink Received Signal Strength Indicator (UL RSSI) during high capacity stadium events. Radial Basis Functions (RBF's) are used to aid in developing the model for the UL RSSI as the amount of users in the stadium changes. In [63] the authors apply distributed optimization dynamics of Mc Culloch Pitts ANN to RAN selection in heterogeneous networks. The algorithm focuses on both network load balancing and end user QoS support. In [59] the authors propose an autonomous optimization method for heterogeneous wireless networks, in which mobile terminals autonomously utilize the most appropriate wireless infrastructure.

A number of ANN approaches are focused on the optimisation of handover without specific regard to network load balancing. Some of these studies exploit the characteristics of ANN in order to investigate the trade-off between excessive spurious handovers and increased handover delay. In [12] the authors propose a fuzzy logic handover algorithm with multiple inputs which limits the potential for spurious handovers. In [13] the authors propose handover management schemes which utilise Fuzzy-Based logic in order to avoid multiple spurious handovers in wireless systems. In [14] the authors evaluate a similar usage scenario to that outlined in this paper; in micro-cellular environments in cities, users often move on predetermined paths. The authors propose to address the trade-off between spurious handovers and handoff delay using a MLP. This approach is similar to FRAME however the focus of the investigation is more limited; defining the optimal handover aggressiveness to mediate between spurious handoff and excessive handover delay. Also the input metrics are limited to RSS.

VIII. CONCLUSIONS

In this work we proposed FRAME, a feed forward neural network-based framework for seamless handover, which uses the predictable nature of vehicle movement to optimise MIH event triggering. FRAME uses a supervised back propagation learning mechanism which captures both cyclical and dynamic performance characteristics. FRAME provides a pluggable extensible interface which can adapt to emerging media stream metrics and device characteristic improvements. In this work, FRAME is evaluated with two forms of learning: frame loss rate directed learning and PSNR directed learning based on deviations between the original and streamed media file. We evaluate FRAME using performance metrics in scenarios deployed on a commercial network in Dublin, Ireland.

The aim of FRAME is to provide a mobile multimedia user with a seamless handover experience. Results illustrate that the more computationally complex PSNR learning method has some improved performance over the frame loss learning approach. However, the selection of an appropriate learning method for FRAME is an implementation specific tradeoff between learning mechanism complexity, particularly for

memory constrained devices, and the level of performance improvement achievable.

Our investigation illustrates that the effectiveness of the FRAME algorithm depends on (1) the availability of candidate APs; where there is a high availability of candidate AP the dynamic probing of network conditions by FRAME gives significant performance improvement over static threshold based MIH event triggering approaches (2) the clustering of weight configurations; a large cluster of weight configurations centered on a suboptimal point will increase training complexity. However, results illustrate that when a large number of weight configurations are clustered around local minima, the performance of FRAME, even during training, significantly exceeds that of traditional static threshold based MIH event triggering approaches.

Future work will focus on (a) the utilization of alternative learning metrics (b) optimization of end user experience by considering device specific performance metrics such as battery life, processor speed and available memory.

REFERENCES

- [1] Institute of Electrical and Electronics Engineers, "IEEE Standard for Local and metropolitan area networks – Part 21: Media Independent Handover Services", 21 Jan. 2009.
- [2] Institute of Electrical and Electronics Engineers, "IEEE Draft Standard for Media Independent Handover Services - Amendment: Handovers with Downlink Only Technologies", 27 Oct 2011
- [3] Institute of Electrical and Electronics Engineers, "IEEE Draft Standard for Local and Metropolitan Area Networks: Media Independent Handover Services - Amendment for Security Extensions to Media Independent Handover Services and Protocol", 29 Nov 2011
- [4] NIST, "The Network Simulator NS-2 NIST Add-on – Neighbor Discovery", January 2007
- [5] M. Li, S. Chen, D. Xie, "A multi-step vertical handoff mechanism for cellular multi-hop networks," in Proc. 2nd ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks, New York, NY, USA, 2007, pp. 119–123.
- [6] B. Ciubotaru, G-M. Muntean, "SASHA—A Quality-Oriented Handover Algorithm for Multimedia Content Delivery to Mobile Users," IEEE Transactions on Broadcasting, vol.55, no.2, pp.437-450, June 2009
- [7] B.-J. Chang, J.-F. Chen, "Cross-layer-based adaptive vertical handoff with predictive RSS in heterogeneous wireless networks." IEEE Transactions on Vehicular Technology, 57 6 (2008), pp. 3679–3692
- [8] L.-H. Chen, G.-H. Chen, M.-H. Jin, E. Wu, "A Novel RSS-Based Indoor Positioning Algorithm Using Mobility Prediction," in Proc. ICPP Workshops 2010: 549-553
- [9] T.-Y. Chung, Y.-M. Chen, P.-C. Mao, C.-K. Tsai, S.-W. Lai, C.-P. Chen, "The Design and Implementation of IEEE 802.21 and Its Application on Wireless VoIP," in Proc. Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st , vol., no., pp.1-5, 16-19 May 2010
- [10] I. Joe, S. Hong, "A mobility-based prediction algorithm for vertical handover in hybrid wireless networks," in Proc. 2nd IEEE/IFIP International Workshop on Broadband Convergence Networks, 2007, pp. 1–5.
- [11] G. Chen, S. Mei, Z. Yong, "Position-prediction-based motion classification assisted strategy in mobility management," in Proc. 7th International Conference on Information,

- Communications and Signal Processing, 2009. ICICS 2009., vol., no., pp.1-5, 8-10 Dec. 2009
- [12] A. Singhrova, N. Prakash, "Adaptive Vertical Handoff Decision Algorithm for Wireless Heterogeneous Networks," in Proc. 11th IEEE International Conference on High Performance Computing and Communications, 2009. HPCC '09., vol., no., pp.476-481, 25-27 June 2009
- [13] G. Mino, L. Barolli, A. Durresi, F. Xhafa, A. Koyama, "A Comparison Study of Two Fuzzy-Based Handover Systems for Avoiding Ping-Pong Effect in Wireless Cellular Networks," in Proc. International Conference on Network-Based Information Systems, 2009. NBIS '09. pp.564-571, 19-21 Aug. 2009.
- [14] S. Yadollah, A. Ghasem, "A pattern recognition based handoff algorithm for micro-cellular systems," in Proc. 19th Iranian Conference on Electrical Engineering (ICEE), 2011, vol., no., pp.1, 17-19 May 2011
- [15] M. Kennedy, H. Venkataraman, G.-M. Muntean, "Battery and Stream-Aware Adaptive Multimedia Delivery for Wireless Devices," in Proc. 7th IEEE International Workshop on Performance and Management of Wireless and Mobile Networks (P2MNET), Denver, USA, October 2010.
- [16] E. Fallon, Additional Testing Results, Available Online, <http://sites.google.com/site/endafallon/ieeetob>
- [17] R. Stewart, Q. Xie et.al. Stream Control Transmission Protocol, RFC 4960, Sep. 2007
- [18] L. Budzisz, R. Ferrus, K. Grinnemo, A. Brunstrom, C. Ferran "An Analytical Estimation of the Failover Time in SCTP Multihoming Scenarios," in Proc. Wireless Communications and Networking Conference (WCNC) 2007
- [19] S. Fallon, P. Jacob, Y. Qiao, A. Hanley, L. Murphy, "Using 802.11 MAC Retransmissions for Path Selection in Multi-Homed Transport Layer Protocols," in Proc. IEEE Global Telecommunications Conference, 2009. GLOBECOM 2009., vol., no., pp.1-6, Nov. 30 2009-Dec. 4 2009
- [20] J. Grossberg "Adaptive Pattern Classification and Universal Recording: I Parallel Development and Coding of Neural Feature Detectors," in Proc. Biological Cybernetics, vol. 23, pp. 121-134, 1976.
- [21] T. Kohonen "Self-organized formation of topologically correct feature maps," Biological Cybernetics Volume 43, Number 1, 59-69, DOI: 10.1007/BF00337288
- [22] W. McCulloch, W Pitts "A logical calculus of the ideas immanent in nervous activity" Bulletin of Mathematical Biology, Volume 5, Number 4 (1943), pp. 115-133
- [23] D. Hebb "The Organization of Behavior: A europsychological Theory", published by John Wiley and Sons, New York, NY 1949
- [24] F. Rosenblatt "A comparison of several perceptron models." in *Self-Organizing Systems – 1962*, Yovits, Jacobi & Goldstein (editors), Spartan Books, 1962
- [25] J. Moody, C. J. Darken, "Fast learning in networks of locally tuned processing units," Neural Computation, 1, 281-294 (1989)
- [26] N. Capela, J. Soares, P. Neves, S. Sargento, "An Architecture for Optimized Inter-Technology Handovers: Experimental Study," Communications (ICC), 2011 IEEE International Conference on , vol., no., pp.1-6, 5-9 June 2011
- [27] S. Fu and M. Atiquzzaman. SIGMA: A Transport Layer Handover Protocol for Mobile Terrestrial and Space Networks. In J. Ascenso, L. Vasiu, C. Belo, and M. Saramago, editors, Invited book chapter in e-Business and Telecommunication Networks, pages 41–52. Springer, 2006
- [28] E. Fallon, L. Murphy, J. Murphy, C. Ma "TRAWL – A Traffic Route Adapted Weighted Learning Algorithm" Proc. 9th International Conference on Wired/Wireless Internet Communications (WWIC 2011) Vilanova i la Geltrú, Catalonia, Spain, , Springer Lecture Notes in Computer Science 6649, pp. 1--14
- [29] "Framework and overall objectives of the future development of IMT-2000 and systems beyond IMT- 2000" Recommendation M.1645
- [30] S. Heikkilä "Business Models of Mobile Operators for WLAN: Case Examples" available for download www.netlab.tkk.fi/opetus/s383042/2006/papers_pdf/G2.pdf
- [31] Netstumbler Network Analysis Software, <http://www.netstumbler.com/>
- [32] Google Earth: Available Online: <http://earth.google.com>
- [33] C.-H. Ke, C.-K. Shieh, W.-S. Hwang, A. Ziviani, "An Evaluation Framework for More Realistic Simulations of MPEG Video Transmission," Journal of Information Science and Engineering
- [34] C. Xu, E. Fallon, Y. Qiao, L. Zhong, G.-M. Muntean, "Performance Evaluation of Multimedia Content Distribution Over Multi-Homed Wireless Networks," IEEE Transactions on Broadcasting Issue 2 June 2011
- [35] C. Palazzi, B. Chin, P. Ray, G. Pau, M. Gerla, M. Rocchetti "High Mobility in a Realistic Wireless Environment: a Mobile IP Handoff Model for NS-2" in Proc. TridentCom 2007.
- [36] I. Lassoued, J.M. Bonnin, B.Z. Hamouda, A. Belghith, "A methodology for evaluating vertical handoff decision mechanisms," in Proc. 7th IEEE International Conference on Networking, 2008, pp. 377–384.
- [37] C. Gu, M. Song, Y. Zhang and J. Song, "Access network selection strategy using position prediction in heterogeneous wireless networks," in Proc. Frontiers of Electrical and Electronic Engineering in China, 5 1 (2010), pp. 23–28
- [38] T. Inzerilli, A.M. Vegni, A. Neri, R. Cusani, "A location-based vertical handover algorithm for limitation of the ping-pong effect," in Proc. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2008, pp. 385–389.
- [39] I. Joe, M.C. Shin, "A Mobility-Based Prediction Algorithm with Dynamic LGD Triggering for Vertical Handover," in Proc. 7th IEEE Consumer Communications and Networking Conference (CCNC), 2010, pp. 1-2
- [40] Y. Y. An, B. H. Yae, K. W. Lee, Y. Z. Cho, and W. Y. Jung, "Reduction of Handover Latency using MIH Services in MIPv6," in Proc. 20th International Conference on Advanced Information Networking and Applications, AINA'06, 2006.
- [41] S. Yoo, D. Cypher, N. Golmie "Timely Effective Handover Mechanism in Heterogeneous Wireless Networks," Journal of Wireless Personal Communications, vol. 55, pp. 449-475, 2008
- [42] A. Mateus, R.-N. Marinheiro, "A Media Independent Information Service Integration Architecture for Media Independent Handover," in Proc. 2010 Ninth International Conference on Networks (ICN), April 2010, pp 173
- [43] P. Neves, F. Fontes, S. Sargento, M. Melo, and K. Pentikousis, "Enhanced Media Independent Handover Framework," in Proc. 69th IEEE Vehicular Technology Conference, VTC Spring 2009., pp. 1-5, April 2009
- [44] B. Sueng Jae Chung, M.-Y. So, "Handover triggering mechanism based on IEEE 802.21 in heterogeneous networks with LTE and WLAN," in Proc. 2011 International Conference on Information Networking (ICOIN), Jan. 2011, pp 399
- [45] M. Xiong, J. Cao, J. Zhang, "Context-Aware Mechanism for IEEE 802.21 Media Independent Handover," in Proc. 20th International Conference on Computer Communications and

- Networks (ICCCN), 2011, vol., no., pp.1-6, July 31 2011-Aug. 4 2011
- [46] K. Kim, Y.-J. Choi, J. Jee, E. Kim, "L2-triggered mobility management in heterogeneous IP wireless networks," in Proc. Third International Conference on Ubiquitous and Future Networks (ICUFN), 2011, vol., no., pp.375-380, 15-17 June 2011
- [47] Y. Kim, S.-K. Lee, "MSCTP-Based Handover Scheme for Vehicular Networks," IEEE Communications Letters, , vol.15, no.8, pp.828-830, August 2011
- [48] Y. Qiao, E. Fallon, J. Murphy, L. Murphy, "SCTP Performance Issue on Path Delay Differential," WWIC 2007
- [49] S. Fallon, P. Jacob, Y. Qiao, L. Murphy, E. Fallon, A. Hanley, "SCTP Switchover Performance Issues in WLAN Environments," in Proc. 5th IEEE Consumer Communications and Networking Conference, 2008. CCNC 2008., vol., no., pp.564-568, 10-12 Jan. 2008
- [50] J.-L. Chen, Y.-W. Ma, Y.-M. Huang, Q.-T. Yang, "An Adaptive QoS mechanism for multimedia applications over next generation vehicular network," in Proc. 5th International ICST Conference on Communications and Networking in China (CHINACOM), 2010, vol., no., pp.1-5, 25-27 Aug. 2010
- [51] E. C. Popovici, T.P. Beganu, M. Gavrilescu, V. Andrei, O. Fratu, S.V. Halunga, "A streaming application for vertical handover testing in wireless hybrid access networks," in Proc. IEEE EUROCON - International Conference on Computer as a Tool (EUROCON), 2011, vol., no., pp.1-4, 27-29 April 2011
- [52] J. Shim, S.-W. Song, B. -H. Rhee, "Enhancement of mSCTP mobility scheme for seamless vertical handover," in proc. Third International Conference on Ubiquitous and Future Networks (ICUFN), 2011, vol., no., pp.268-272, 15-17 June 2011
- [53] J. Baek, P.S. Fisher, J. Minho, "An Enhancement of mSCTP Handover with an Adaptive Primary Path Switching Scheme," in Proc. 72nd IEEE Vehicular Technology Conference Fall (VTC 2010-Fall), 2010, vol., no., pp.1-5, 6-9 Sept. 2010
- [54] T. Ming, Y. Hewei, "A smooth handoff scheme based on mSCTP for speed-aware application in mobile IPV6," in Proc. 2nd IEEE International Conference on Broadband Network & Multimedia Technology, 2009. IC-BNMT '09, vol., no., pp.586-591, 18-20 Oct. 2009
- [56] A.-J. Höglund, K. Hätönen, "Computer Network User Behavior Visualization using Self-Organizing Maps," in Proc. IEEE International Conference on Artificial Neural Networks (ICANN), vol. 2, 1998, pp. 899-904.
- [57] A.-J. Höglund, K. Hätönen, A.-S. Sorvari, "A computer host-based user anomaly detection system using the self-organizing map," in Proc. IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN), vol. 5, pp. 411-416. 2000 IEEE
- [58] J. Laiho, M. Kylväjä, A.-J. Höglund, 2002, "Utilization of Advanced Analysis Methods in UMTS Networks," in Proc. 55th IEEE Vehicular Technology Conference (VTC) Spring, vol. 2, pp. 726 -730. 2002
- [59] M. Hasegawa, K. Ishizu, H. Murakami, H. Harada "Experimental evaluation of distributed radio resource optimization algorithm based on the neural networks for Cognitive Wireless Cloud," in Proc. IEEE PIMRC Workshops, pp. 32-37 2010
- [60] M. Hasegawa et.al "Design and Implementation of A Distributed Radio Resource Usage Optimization Algorithm for Heterogeneous Wireless Networks," in Proc. IEEE Vehicular Technology Conference (VTC) Fall, pp. 1-7 2009
- [61] F. Yuang, N. Xiong, J.-H. Park "A self-adaptive K selection mechanism for re-authentication load balancing in large-scale systems," Journal of Super Computing 15th July 2011
- [62] S. Roets, P. Reddy, P. Govender "UL RSSI as a design consideration for distributed antenna systems, using a radial basis function model for UL RSSI," in Proc. the 5th international conference on Network optimization Inoc 2011
- [63] M. Hasegawa, K. Ishizu, H. Murakami, H. Harada, "Experimental evaluation of distributed radio resource optimization algorithm based on the neural networks for Cognitive Wireless Cloud," in Proc. 21st IEEE International Symposium on Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops), 2010, vol., no., pp.32-37, 26-30 Sept. 2010
- [64] K.R. Kumar, P. Angolkar, D. Das, "SWiFT: A Novel Architecture for Seamless Wireless Internet for Fast Trains," Ramalingam, R. VTC Spring 2008. IEEE, pp. 3011-3015, May 2008



Enda Fallon received a B.Sc. in computer science and mathematics from University College Galway and an M.Sc. in software engineering from Athlone Institute of Technology. He is currently working towards a Ph.D. at the Performance Engineering Laboratory at University College Dublin.

He joined Athlone Institute of Technology (AIT) from Ericsson in 2002. In 2003, he founded the Software Research Institute (SRI) at AIT. Since 2003, he has been principal investigator on over 25 collaborative industry/academic research projects. He has published over 40 papers in top-level international conferences and journals. His research interest focuses on service mediation and adaptation for heterogeneous networking environments.



Liam Murphy received a B.E. in Electrical Engineering from University College Dublin in 1985, & an M.Sc. and Ph.D. in Electrical Engineering and Computer Sciences from the University of California, Berkeley in 1988 and 1992 respectively. He is currently a Professor of Computer Science & Informatics at University College Dublin, where he is a director of the Performance Engineering Laboratory.

Prof. Murphy has published almost 150 refereed journal and conference papers on various topics, including multimedia transmissions, dynamic and adaptive resource allocation algorithms in computer/communication networks, and software development. His current research projects involve computer network convergence and software performance engineering. Prof. Murphy is a Member of the IEEE (Communications, Broadcasting, and Computer societies) and a Fellow of the Irish Computer Society.



John Murphy is an Associate Professor in Computer Science and Informatics at University College Dublin. He got a first class honours degree in electronic engineering (B.E.) in 1988 from the National University of Ireland (UCD), an M.Sc. in electrical engineering from the California Institute of Technology in 1990 and a Ph.D. in electronic engineering from Dublin City University in March 1996.

He is an IBM Faculty Fellow, a Senior Member of the IEEE, a Fellow and Chartered Engineer with Engineers Ireland, and a Fellow of the Irish Computer Society. For many years he held an academic part-time position at the Jet Propulsion Laboratory in Pasadena, and acted as a consultant to the US Department of Justice. Prof. Murphy is an associate editor for IEEE

Communications Letters Journal and an associate editor for Telecommunications Systems Journal. He was the guest editor (along with Prof. Perros) for a special issue of IET Communications on Optical Burst and Packet Switching in 2009. He has published over 100 peer-reviewed journal articles or international conference full papers in performance engineering of networks and distributed systems and has been awarded over 20 competitive research grants (over 5 million euro). He has supervised 13 Ph.D. students to completion.



Gabriel-Miro Muntean (S'02–M'04) received the B.Eng. and M.Sc. degrees in software engineering from the Computer Science Department, “Politehnica” University of Timisoara, Romania in 1996 and 1997; and the Ph.D. degree from Dublin City University, Ireland for research in the area of quality-oriented adaptive multimedia streaming in 2003.

He is a Lecturer with the School of Electronic Engineering and co-Director of the Performance Engineering Laboratory at Dublin City University, Ireland. He has published over 120 papers in top-level international conferences and journals and has authored a book and ten book chapters and edited four books. His research interests include quality and performance-related issues of adaptive multimedia delivery, and personalized e-learning over wired and wireless networks and with various devices. Dr. Muntean is Associate Editor with the IEEE TRANSACTIONS ON BROADCASTING, Associate Editor with the IEEE COMMUNICATIONS SURVEY AND TUTORIALS and reviewer for important international journals, conferences and funding agencies. He is member of IEEE, IEEE Broadcast Technology Society, *lero* – the Irish Software Engineering Research Centre and the *Rince* Research Institute Ireland.