Avoiding handover interruptions in pervasive communication applications through machine learning

Vijaya Nirmala Mitnala
University of Essex
Colchester, UK CO4 3SQ
vm20821@essex.ac.uk

Martin J. Reed
University of Essex
Colchester, UK CO4 3SQ
mjreed@essex.ac.uk

Ian Kegel
British Telecommunications plc. (BT)
Adastral Park, Ipswich, UK IP5 3RE
ian.c.kegel@bt.com

John Bicknell
British Telecommunications plc. (BT)
Adastral Park, Ipswich, UK IP5 3RE
john.bicknell@bt.com

Abstract—**There are an increasing diversity of devices and networks for making and receiving video and/or audio calls. The prevalence of smartphones, multiple smart speakers and various computing platforms in many homes has led to much greater flexibility in how and where users communicate. This trend is only set to continue as AR/VR becomes more commonplace and services such as healthcare, for which flexible, high-quality communications is a critical component, move online. Despite these trends, comparatively little has been done to offer a converged communications experience once a speech call session is in progress. For example, the simple act of switching an ongoing call from a smartphone to a smart speaker is often a highly manual process. Pervasive communication systems aim to address this by providing a seamless, flexible communications experience across multiple devices and, where required, multiple networks. A key part of achieving this experience is through seamless session handover. This paper considers how session initiation protocol systems can operate in a pervasive communication scenario, in particular when there is mobility causing vertical handover between delivery networks. The paper uses a machine learning-based approach to predict user's transitions and thus overcome interruptions in speech due to signalling handover. As pervasive communication systems are not currently available for measuring the performance of the solution, the paper uses a commonly available dataset for vehicular mobility to assess likely handover performance. The results show that prediction can reduce the more concerning interruptions (>2s) due to vertical handover events by more than 99.9%.**

Key words– Heterogeneous communication systems, Pervasive communication systems, Supervised Machine learning, Session Initiation Protocol, Vertical Handover.

I. INTRODUCTION

Pervasive communication aims to deliver a seamless audio and/or video experience based on situational context – such as the preferred end device or best available network – both before and during a call session. To do this, pervasive communication systems must be able to provide seamless transitions between different access and backhaul networks when required (such as 5G and 4G RAN and private or public Wi-Fi with broadband backhaul) and different audio presentation systems (such as one or more smart speakers, smartphones, tablets and PCs). Despite the growth of IP telephony in recent years, and the diversity of devices and networks on which it is now used, support for pervasive communications remains limited. Mobile network standards [1] have recognised the importance of IP mobility management mechanisms to provide the best chance of a seamless transition between network types, but do not consider transitions between devices. Smart speaker providers have implemented 'group chat' functionality [2] so that communication sessions can be shared across multiple devices, but do not provide the ability for a call session to be transferred between devices.

Seamless session handover is therefore one of the key features of pervasive communication systems. In this work we consider how the session initiation protocol (SIP) can be used in IP telephony systems where a seamless transition is required. In the absence of an existing pervasive communication system we consider how SIP sessions would transition in a worse-case scenario where vertical handovers are required. This scenario is closely related to various real-world use cases - such as a 5G or 4G cellular call transitioning to a private or public Wi-Fi network – but it also serves as a useful limiting case for the pervasive communication scenario where it is envisaged that temporal disruption to the audio stream will be a significant factor affecting user experience.

Considering the scope of pervasive communication system for transitioning a speech call between devices and also the worst-case scenario of between the networks, for example moving between Wi-Fi hotspots, this paper is scoped for optimising the session handling between the networks. And hence the main contributions of this paper are towards understanding likely call interruptions for SIP based speech when transitioning between heterogeneous networks and improving on this problem using a Machine learning (ML) approach to provide session continuity with no, or little interruption. The paper uses a data-driven approach by using existing mobility data from a real-world vehicular mobility dataset to predict next network transitions. Using this prediction, speech can be transmitted in advance to allow a *make-before-break* handover in heterogeneous networks in a manner that we are familiar with from homogeneous networks, such as LTE. While the paper uses a vehicular dataset for evaluating the problem and solution, in the absence of existing real-world pervasive communication application data, the approach is designed to operate across any type of IP networks where IP connectivity may change during the conversation.

In II we review related literature before describing our proposed methods for both SIP and ML in III. Our results in IV show the benefit of the proposal which we conclude in V.

II. Related Work

The Session Initiation Protocol (SIP) [3] has been widely been used as an application layer signalling protocol in VoIP and IMS systems [4]. Several studies analyzed the session interruption using SIP especially for *mid-call* terminal mobility [5] and improving interruption using proactive handover schemes with SIP extensions [6] [7]. A handover delay comparison study in vertical handover with SIP and without SIP is carried in [8]. Adapting Machine learning (ML) techniques in SIP have recently emerged as an alternative to traditional models to improve QoS in SIP. For example recent work describes using ML to detect attacks against SIP-based services [9] [10] [11]. However, in this paper we are concerned with alternative uses of ML for predicting handover.

Although the authors are not aware of any papers directly addressing session handover for SIP using ML, using ML-based user movement prediction in vehicular communication and networks has been studied previously. In particular, it has been widely studied for the purpose of effective routing towards achieving seamless traffic flow. The work in [12] propose a Supervised ML approach to predict cell and link association duration using cell position and cell load. The motivation for carrying the work is for anticipating network control decisions in Software Defined Vehicular Networks. But the paper does not describe how these predictions could provide the control decisions. Consequently, our paper describes how the handover interruptions can be reduced using the network transition predictions. The work in [13] details an ML approach to predict the handover in Heterogeneous networks in an IoT environment. The ML predictive analysis over downlink throughput in [14] for Mobile Broadband networks has gained high QoS for end users in video streaming.

The works above differ from the approach in this paper in that they approach either only position information or addressing routing issues, whereas this paper specifically aims at solving the session transition which although related, has important differences. Specifically, the lightweight approach of proactive handover using ML techniques (software controlled movement prediction and predicting the future attachment point before movement) at application layer has advantage over traditional network break-before-make or so called make-before-break approaches [7]. This is because the former can predict the handover in the manner required for the application while the later is totally dependant on updates from link-layer for performing handover without understanding the application layer needs. Additionally, modern ML can bring many advantages in the processing of bulk data from various network sources and even with a large number handovers. Consequently this paper shows that using ML for improving handovers and QoS in SIP can bring a greater evolution in telecommunication systems.

III. Proposed Method

The subject of proactive vertical handover using SIP has already been studied for some time in the past. However the techniques for proactive handover were based on soft handover ("make before break") in link-layer in homogeneous networks [7]. In our study, we are considering the handover execution at application layer using SIP and the handover decision is supported through ML processing by predicting the next likely, attachment point and, optionally, the estimated time to move to this next attachment point. It should be noted here that the bulk data processing power of ML is carried out at the training stage and that the processing at the prediction stage is relatively simple and this is helpful in handling a large number of handovers. The ML is carried out in a central server for the proposed pervasive communication SIP application. A suitable server is the SIP Registration Server [15] which receives updates with user registrations and change of point of attachment. This needs to identify only the local attachment network that a device is connected to, for example different WiFi hot-spots; crucially, the ML does not need actual geographical location data. While geographical data maybe available in some horizontal handover systems, such as LTE/5G, this information is not generally available in heterogeneous networks which is the focus here. In this paper we are using a Supervised learning approach as we are using an existing dataset *i.e.*, mobility data learned from a previous day with *ground-truth* labels. In practice it is more likely to be implemented as a semi-supervised algorithm with the model refined on a daily or weekly basis from past data. The ML uses information about attachment networks and times between changes of attachment to predict the next likely network attachment and the time to change to this new attachment. Thus, SIP session continuity can now be provided through the prediction from the ML.

The approach of ML prediction for handing over the session can be extended in a pervasive communication system where handover is between the devices. This paper uses the history of point of network attachment and duration of attachment to predict the network handover. These features can be applied for device attachment prediction with some modification, specifically, we adapt the handling the SIP session by using an out of band mechanism such as a new pervasive app running on the devices could be used for generating the features to predict the device attachment. After a session is predicted to soon transition to a new device, seamless SIP session handover can happen in similar manner to the IP address change or by conference setup as described below.

A. SIP session continuity

We have given, above, a brief high-level overview of the mechanism proposed in this paper, here we discuss one possible implementation of the mechanism in the context of the SIP signalling. Vertical handover can occur during *pre-call* or during *mid-call*. This paper considers the *mid-call* mobility scenario, where session and media update is expected for the existing session.

When the mobile node moves from one network to another

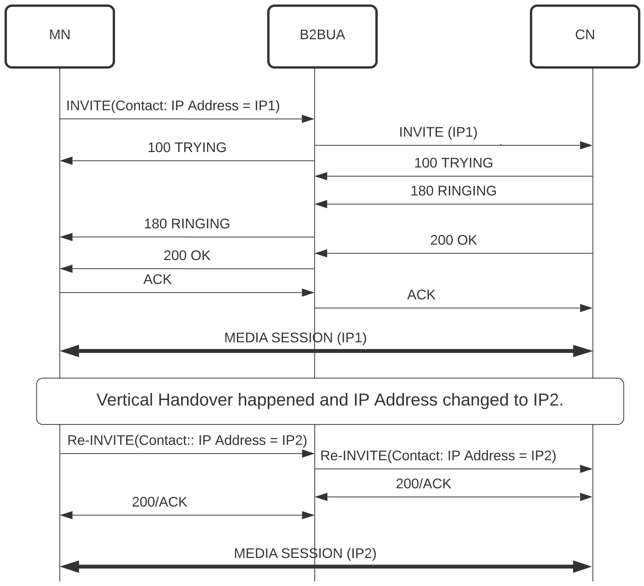

Fig. 1. SIP mid-call flow when there is an address change due to handover between a mobile node (MN) and a corresponding node (CN)

it first contacts the Registration Server[1] to de-register the previous IP address and register the new address at the new point of attachment. When the Registration Server coordinates with the SIP Location Server to register this new address it queries the ML predictor using: the previous point of attachment, the new point of attachment and the time interval between registration of the previous and new point of attachment. From this query, the ML predictor then estimates the next point of attachment and, optionally, an estimate of the time to change to the next point of attachment. Consequently, an agent at the new attachment point is notified to expect a likely new attachment in the future and replies back to the Registration Server with the future IP address. Finally, the mobile node is notified, via a SIP NOTIFY message, of the next likely point of attachment.

The existing SIP signalling dialog for a *mid-call* change of network attachment is shown in Fig 1. It sends a re-INVITE to its corresponding node with its new IP address as its new Contact [16]. There are challenges using re-INVITE in the existing dialog, due to the handover delay, SIP re-session overhead, Registration/Location overhead and SIP message size; this is not helped by the text-based nature of SIP. Crucially, media handover cannot happen until the signalling messages are complete and consequently there are interruptions to the speech during this handover when using existing SIP-based mechanism. Consequently, this paper will propose modifications to the signalling process.

In this paper it is proposed that the media handover interruption is removed using proactive media session handling using the previously described ML to predict the network

[1]Note, we use a singular form for a registration server; however, in practice multiple actual servers will be used, in this case the daily updated ML model would need to be synchronised across the registration servers.

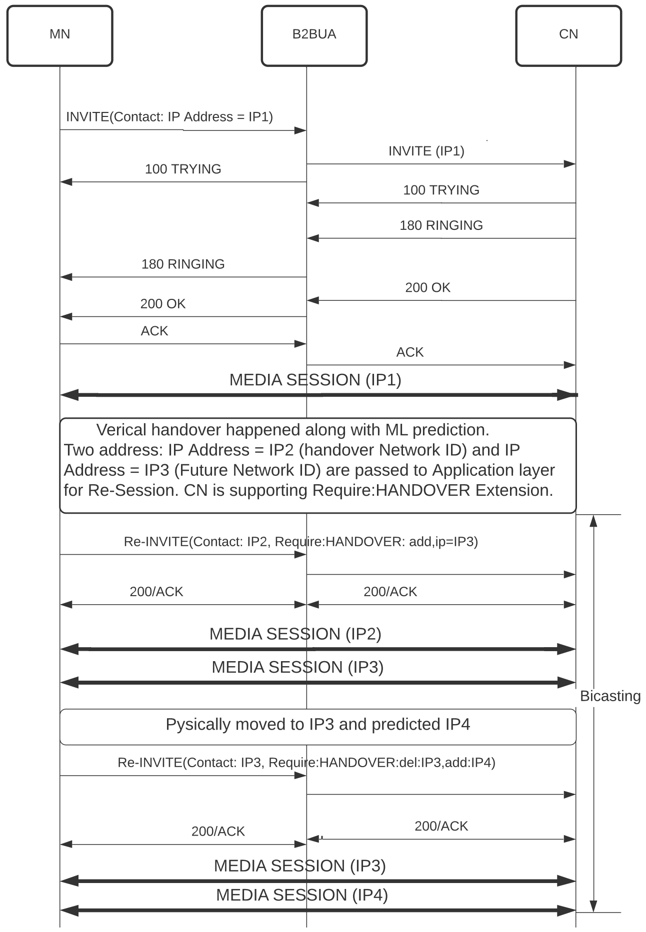

Fig. 2. SIP mid-call flow with Two IP Addresses - BiCasting: Using SIP Extension

mobility so that media delivery is established before the physical move *i.e.*, *"make before move*." The existing SIP standards [16] do not support, session and media on two IP addresses. However a session handover approach called *bicasting* using a SIP Extension is proposed by Toktam *et al.* [7] in the context of session setup using *"make before break"*. While bicasting does not solve all the problems in a heterogeneous network, as it was designed to operate on the Link-layer only, the signalling and media delivery can be re-utilised in this paper. Specifically, bicasting is used to maintain media continuity by sending media to both the current and the next predicted point of attachment. The session call flow using bicasting, when the corresponding node supports the Require:HANDOVER extension, is as shown in Fig 2. In the case that the corresponding node does not support the Require:HANDOVER extension, then solutions like a back-to-back user agent (B2BUA) can be used as a bridge for both session and media as proposed in [6] for backward compatibility.

B. Mobility dataset

To analyse our mobility prediction for a pervasive communication application we would ideally need realistic data from a communication application operating over heterogeneous networks, for example moving between Wi-Fi hotspots or between a mobile phone and a smart speaker. In the absence of realistic data from such a future application we have used a vehicular mobility dataset. The main difference between the vehicular mobility dataset and the expected pervasive communication application is that the former deals with a greater number of networks with fewer handovers while the pervasive communication application is expected to deal with only one or two networks (e.g. between a smart speaker and a phone) but more frequent handovers.

The handover prediction example used for this paper is based upon actual mobility data from the city of Cologne (described in more detail in III-B1 below). Paired with this we used actual network locations (again III-B2 below) from Cologne to simulate where these actual users were connected as they moved in their vehicles. While it would be possible, from this first dataset, to use the vehicle location for the purpose of prediction, this is unrealistic in a practical networked application scenario as network elements do not normally have access to this information. Consequently, for the ML prediction we used only the previous network attachment identifier (ID), the current network attachment ID and the duration in the previous network. This was then used to predict the next network ID and duration at the current network ID *i.e.,* the time until the next transition. These features were determined from a second network attachment dataset that was generated from the vehicular mobility dataset as described below.

1) Vehicular Mobility dataset : The mobility trace contains vehicle mobility information with a 1-second granularity for an anonymised vehicle identifier, its position on the two-dimensional plane (X and Y coordinates in meters) and its speed (in meters per second) taken ove a 24 hour duration in the greater urban area of the city of Cologne. This trace contains the mobility behaviour of more than 650,000 vehicles of different types. The generation of this realistic data set is clearly described in research papers associated with the dataset [17] [18]. The parameters of this feature set are described in Table I.

2) Network Attachment dataset: As noted earlier, it is not realistic for network elements to know precise MN locations and speeds. Consequently, to generate the features for the ML we assume only knowledge of network attachment. To simulate this for a future pervasive communication application we used the base station location information from public German databases in 2012 using the same coordinate system as the vehicle positions. Applying a Voronoi Tessellation on base station locations provides a good estimate for each network attachment point in the region [19]. Using this approach we generated the network attachment dataset directly from the realistic vehicular mobility dataset. This resulted in the features shown in Table II.

TABLE I
VEHICULAR MOBILITY DATASET PARAMETERS

Parameter	Description
Vehicle Id	A unique vehicle identifier
Vehicle position	X,Y coordinates of the vehicle position, used for deriving the vehicle's association to nearest wireless access point
Wireless access point position	X,Y coordinates, that can be used to represent a network attachment ID
First TimeStamp	Time of the vehicle joining a new network attachment
Second TimeStamp	Time of the vehicle leaving a network attachment

TABLE II
ML FEATURES GENERATED FROM VEHICULAR MOBILITY DATASET PARAMETERS SHOWN IN TABLE I

Feature	Description
Previous network ID	Previous network attachment ID
Previous attachment duration	Duration that a device was connected with the previous network
Current network ID	Current network attachment ID
Future network ID	The next predicted (or actual) network attachment ID
Current network Duration	Duration predicted (or actual) that this MN will be connected with the current network before moving to the Future network ID

C. Machine learning approach

There are many machine learning algorithms and approaches, and detailed descriptions are given widely in the literature [20] [21]. Analysis of the problem described in this paper means that a supervised (or rather semi-supervised) approach is appropriate; a comparative survey specially on Supervised ML algorithms is given by [22] [23]. This work compared a number of Supervised machine learning techniques [22] including: linear regression, naive Bayes, decision trees, multi-layer neural networks, random forest (RF) and k-nearest neighbour (KNN). After comparison, RF [24] and KNN [25] were found to perform the best and are the only two shown in the paper. While comparing in detail the reasons for these algorithms success is beyond the scope of this paper, it can be noted that the problem is highly non-linear (mapping arbitrary network ID histories); consequently rule based approaches like random forests can perform well in these cases. Additionally, random forests supports both classification and regression from the same input dataset. It was noted that random forests can handle large datasets efficiently, although this was less critical for this application. As noted at the start of III the ML would be implemented as a semi-supervised approach in practice by using data from a previous day; or, day of the week, as mobility patterns vary weekly.

IV. RESULTS

To evaluate the performance of the proposed handover solution, the network attachment dataset was generated as described in III-B. The key metrics of the work are to analyse the call interruption time caused by SIP session handovers. However, a number of options were considered in the application of the ML prediction and these have various impacts in the performance of the proposal in this paper. The results were generated according to these following scenarios:

no prediction (NP) the default using standard SIP handover with no bicasting;

next network prediction (1P) bicasting speech to both the current network and the predicted next network

next network prediction to two most likely (2P) bicasting (or rather "tricasting") to the current network and the two most likely predicted next networks

next network prediction with regressor (Px) send speech to the current network and then bicast to the predicted network x s before the regressor predicts the transition

In the above, the regressor was introduced in the Px scenarios to reduce the time that duplicate traffic is sent to the next network. In practice we will see that, while the regressor has good median prediction, it has a relatively wide variation around this median performance which means that if the prediction is too long there will be an interruption in the call as the transition happened earlier than the predicted transition. Consequently, an error margin x was added to the regressor predictor and these are shown in the results as Px between 2 and 32 seconds.

A. Calculate the SIP session establishment time

We have performed measurements of SIP call session establishment latency using a SIP test-bed under low load conditions, running on a local system *i.e.* such that additional processes, link latencies or CPU limitations were not significant. The session successful establishment latency for a single user repeated 100 times are shown in Fig. 3 indicating that median times for call establishment (or re-establishment after handover) are in the order of 280 ms. In practice there may be additional factors that will increase this latency namely, additional proxies, re-registration process, load on the proxies, distance from user to proxies *etc*. Our measurement of 280 ms is comparable with others that have measured SIP signalling latency [26]. Also, we are assuming this session set up time includes time required for re-registration and location updates, again in practice this might cause higher latency. For the modelling results, we sampled following the measured latency distributions shown in Fig 3 so that the results were based upon these real latency values.

B. Calculation of Network Attachment dataset

We have analysed handover from approximately 650,000 vehicles across a total of 4.5 million handovers from one network attachment point to another attachment point. A vehicle having at least 2 handovers was considered for processing *i.e.,* static “mobile” nodes were excluded. We assumed each handover is a vertical handover. While the later may not be realistic for a true vehicular case it acts as a useful proxy of generated events for a future pervasive communication application. In order to derive the vehicular transitions, we have created a processing algorithm with inputs: the generated network IDs from Network Attachment dataset, vehicle ID, and timestamps of transitions to determine network attachment duration. The

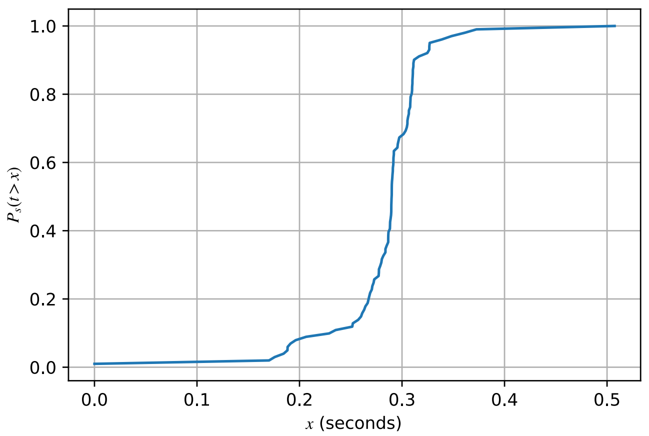

Fig. 3. Empirical cumulative distribution function showing SIP call set up latency t

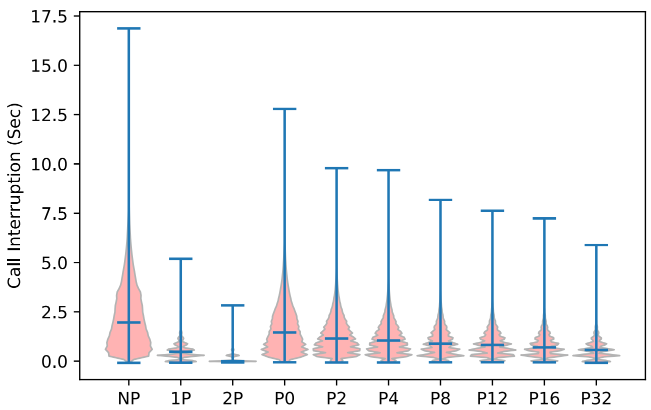

Fig. 4. Call interruptions for the cases of no prediction (NP), using one (1P) or two (2P) previous network IDs, and using the regressor to predict transition time (Px) with varying regressor error margins x

vehicle’s association with a nearest network attachment point was determined from the vehicular mobility dataset using the k-d Tree algorithm [27].

C. ML performance

In order to predict the next, likely, attachment point and the estimated time to move to this next attachment point, we have trained our model with the history of vehicle transition

TABLE III
PERFORMANCE OF THE CLASSIFIER

	1 prev. net.		2 prev. net.	
Metric	RF	KNN	RF	KNN
Classifier Accuracy	0.793	0.785	0.937	0.929
Classifier F1 score (macro)	0.747	0.737	0.904	0.907
Classifier F1 score (micro)	0.793	0.785	0.937	0.929
Classifier F1 score (weighted)	0.790	0.782	0.936	0.938

TABLE IV
PERFORMANCE OF THE REGRESSOR

	RF	KNN
Regressor Median Absolute Error	7.80	8.049
Regressor Median Relative Error	1.8%	1.9%
Regressor Coefficient of determination R^2	0.299	0.296

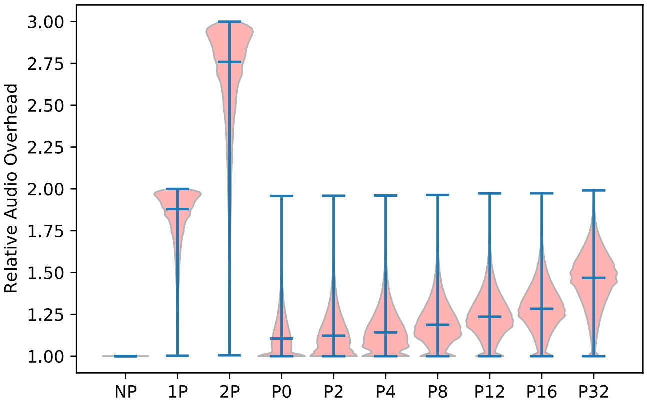

Fig. 5. Relative Audio overhead for the cases of no prediction (NP), using one (1P) or two (2P) previous network IDs, and using the regressor to predict transition time (Px) with varying regressor error margins x

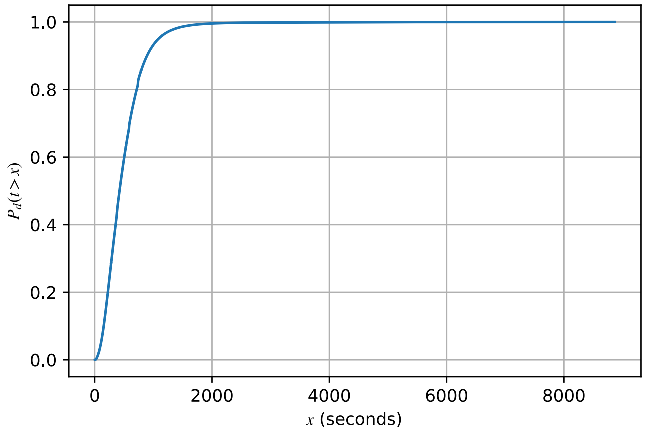

Fig. 6. Empirical cumulative distribution function showing duration of network association t

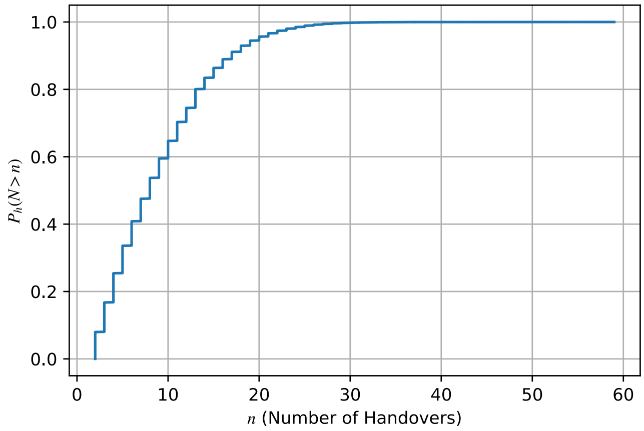

Fig. 7. Empirical cumulative distribution showing number of handovers n for a call

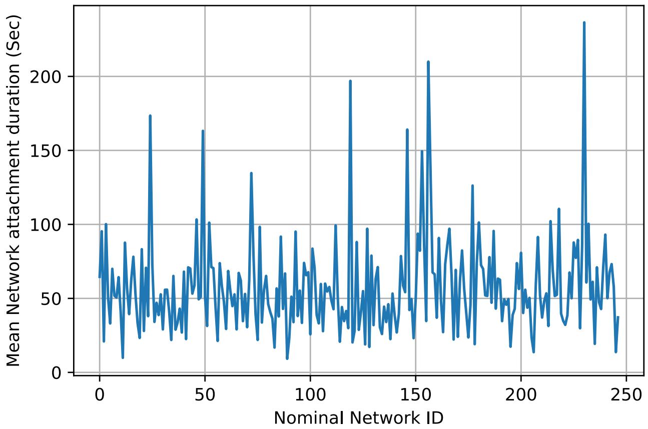

Fig. 8. Mean network duration across the network IDs

i.e. previous network ID, duration in the previous network and current network ID. The estimated time to move to the next attachment point is predicted with RF and KNN regressor algorithms and the next likely the network attachment point prediction is using a RF and KNN classifier algorithms with performance results shown in Tables III and IV. Considering the nature of our model as a *multi-classifier*, we have calculated both the classifier accuracy: representing the ratio of correct predictions to total predictions and F1 scores (*macro, micro and weighted)*, representing the scores of precision and recall. For the regressor we show median absolute and relative error and the coefficient of determination, R^2, showing the performance of the regressor model to predict the duration of

TABLE V
NUMBER OF CALLS WITH INTERRUPTIONS GREATER THAN 300MSEC, 1SEC AND 2SEC RESPECTIVELY WHEN CONSIDERING THE SOLUTIONS WITH NO PREDICTION (NP), PREDICTION FROM ONE PREVIOUS NETWORK (1P), PREDICTION FROM TWO PREVIOUS NETWORKS (2P) AND PREDICTION WITH REGRESSOR AND x SECOND ERROR MARGIN (Px).

Scenario	300 ms	1 s	2 s
NP	**598,518**	**468,029**	**309,945**
1P	398,796	73,950	3,069
2P	**145,241**	**2,304**	**15**
P0	575,124	408,791	222,503
P2	563,410	352,343	135,022
P4	549,849	323,300	10,0745
P8	530,794	277,188	63,414
P12	513,637	235,098	39,733
P16	497,017	200,523	25,801
P32	444,940	118,136	7,647

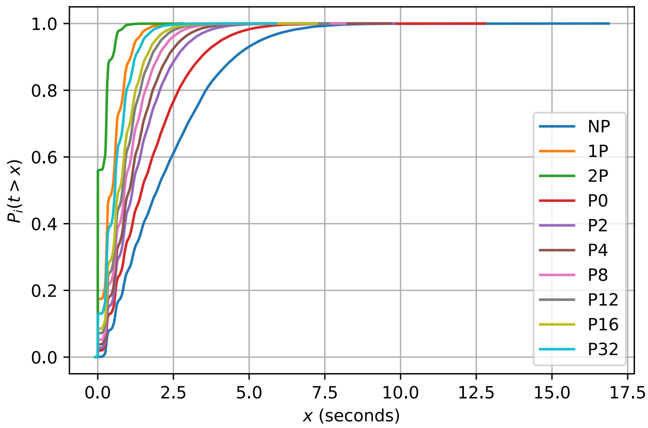

Fig. 9. Empirical cumulative distribution showing call interruptions t

attachment:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})}{\sum(y_i - \bar{y})} \quad (1)$$

a value closer to 1 is considered as a good prediction model. We see from Table IV that the regressor performed very well in terms of median relative error (less than 2%) however with R^2 being low there was a relatively large variation of error around this median value, hence the need for the error margin added to the regressor predictor. We ran our ML model on a single core server and it took approximately 18 hours for 4,500,000 transactions i.e. 70 ML points a second. The statistics of vehicular transitions indicate that median transition rate is one transition every 500 s for each vehicle; i.e. a single core can handle 35,000 simultaneous calls. Considering a typical European Nation like the UK may have approximately 1.5 million simultaneous calls[2] in a mobile network this means that less than 40 cores would be needed for a nationwide prediction service using this ML solution.

D. Call interruption results

After using the above to model the realistic call interruption we can determine how many calls have interruptions above a certain threshold as shown in Table V. Unlike packet loss in VoIP, there is little evidence to support what is an acceptable call interruption time. We have chosen three values from a level that is likely to be just perceptible in many conversations (300 ms, comparable to average word length) to a level of 2 s where it is a considerable portion of a typical sentence. The results in Table V indicate that call interruptions in the case of the longer interruptions of 2 s can be reduced by more than 99.9%, by using the case where two previous network attachment points are used in the ML (2P). Further detail on the call interruption times can be seen in Fig. 4 showing that the prediction with two target network attachments (2P) gives substantially lower call interruption. It can further be seen that using the regressor for predicting the transition without an error margin (P0) does not achieve satisfactory reduction but that adding 32 s (P32) approaches the performance of the 1P case.

One of the costs of the mechanism proposed in this paper is that there is additional traffic in the network during the bicasting. For the cases without the regressor (1P, 2P) this is when a mobile node moves to a new node. The overhead is shown in Fig. 5 using the existing SIP handover without prediction (NP) as the baseline. For the case where bicasting is to one node (1P) or two nodes (2P), there is up to 2 times (1P) and 3 times (2P) the amount of traffic. The reason for double and thrice the amount of traffic is due to upfront media transmission on these predicted network attachment points during the handover. The overall distribution of handovers across the network attachment points is as shown in Fig 7. The transition through the network attachment points is also not uniform as shown in Fig 8 showing that for some network locations, mobile nodes are attached to that network for a much longer time than other network attachment points; this is caused by the diverse nature of network density across the geographical area with high density in the city centre and many transitions and much larger areas in the outer city zones with fewer transitions.

[2] Estimated using data from Ofcom in the UK [28]

In practice a deployer of this solution would need to trade off the overhead shown in Fig 5 and the considerable benefit shown in Fig. 9 and Table V. Some further insight is given in Fig. 9 which shows the probability of call interruptions for the various options. It can be seen that there is substantial benefit by deploying the ML solution in this paper. While the exact trade-off would be a deployment decision, the authors suggest that using the classifier with the regressor, using a suitable error margin, gives an optimal performance. Indeed using the regressor with 32 s error margin (P32) gives performance close to using the classifier alone (1P) but with substantially less transmission overhead by selectively bicasting.

V. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed a machine learning approach to avoid the handover interruptions during the session updates in application layer for pervasive communication applications. A proactive handover *"make before move"* with bicasting of media sessions on two network attachment points; actual and predicted is proposed. Due to unavailability of pervasive communication application transition data, a readily available user vehicular transition dataset is used for the handover analysis. This work compared a number of Supervised machine learning techniques and the experimental results indicate that the random forests can perform well in these types of features due to non-linear nature of the dataset. Mean call interruption is calculated on no prediction and prediction cases and results are plotted. The analysis shows that the more concerning call interruptions can be reduced by more than 99.9% with a modest communication transmission overhead. Our experiments conclude that the overhead of running the ML for handover prediction is small and that a single core running can serve the peak load of approximately 35,000 simultaneous pervasive communication sessions.

Building on this work, it would be valuable to carry out subjective studies of the acceptable limits of temporal disturbance during handover. These limits could potentially be expressed in a similar manner to the impact of one-way delay in the ITU-T G.114 [29] specification. Furthermore, future work could also consider how vertical handover between devices could be optimised in a far-field pervasive communication environment. In the future we plan to develop the pervasive communication system for the scope of seamless session handover between the devices in the personal or business area networks. This will use the same ML process for device prediction and using existing or extended SIP support for session handling. The solution provided in the paper for proactive handover for a device between the networks along with future work providing proactive handover between the devices will provide the complete handover solution for pervasive communication system.

References

[1] 3GPP, "Architecture enhancements for non-3GPP accesses," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.402, June 2019.

[2] M. Humphries. (2020) Amazon adds group chat to echo devices. [Online]. Available: https://www.pcmag.com/news/amazon-adds-group-chat-to-echo-devices

[3] A. B.Johnston, "[]_SIP_Understanding_the_Session_(BookFi.org).pdf," 2004.

[4] M. A. Qadeer, A. H. Khan, J. A. Ansari, and S. Waheed, "IMS network architecture," *Proc. - 2009 Int. Conf. Futur. Comput. Commun. ICFCC 2009*, pp. 329–333, 2009.

[5] R. Mahmood and M. A. Azad, "Sip messages delay analysis in heterogeneous network," in *2010 International Conference on Wireless Communication and Sensor Computing (ICWCSC)*, 2010, pp. 1–5.

[6] E. S. Boysen and J. Flathagen, "Using sip for seamless handover in heterogeneous networks," in *2011 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2011, pp. 1–8.

[7] T. Hemmati and M. H. Y. Moghadam, "Sip-based vertical handover scheme with bicasting," in *16th International Conference on Advanced Communication Technology*, 2014, pp. 19–22.

[8] J. C. Vijayshree and T. G. Palanivelu, "Vertical handover triggering between WLAN and WIMAX using sip," *Proc. 2014 IEEE Int. Conf. Adv. Commun. Control Comput. Technol. ICACCCT 2014*, no. 978, pp. 769–774, 2015.

[9] Z. Tsiatsikas, A. Fakis, D. Papamartzivanos, D. Geneiatakis, G. Kambourakis, and C. Kolias, "Battling against ddos in sip: Is machine learning-based detection an effective weapon?" in *2015 12th International Joint Conference on e-Business and Telecommunications (ICETE)*, vol. 04, 2015, pp. 301–308.

[10] K. Umoto, S. Ata, Y. Chimura, N. Nakamura, and T. Yao, "An automatic error identification method in call control protocol using levenshtein distance," in *2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 2020, pp. 266–273.

[11] D. Pereira, R. Oliveira, and H. S. Kim, "A machine learning approach for prediction of signaling sip dialogs," *IEEE Access*, vol. 9, pp. 44 094–44 106, 2021.

[12] S. Toufga, S. Abdellatif, P. Owezarski, T. Villemur, and D. Relizani, "Effective prediction of v2i link lifetime and vehicle's next cell for software defined vehicular networks: A machine learning approach," in *2019 IEEE Vehicular Networking Conference (VNC)*, 2019, pp. 1–8.

[13] N. Nayakwadi and R. Fatima, "Machine learning based handover execution algorithm for heterogeneous wireless networks," in *2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, 2020, pp. 54–58.

[14] K. Kousias, Ö. Alay, A. Argyriou, A. Lutu, and M. Riegler, "Estimating downlink throughput from end-user measurements in mobile broadband networks," in *2019 IEEE 20th Int. Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2019, pp. 1–10.

[15] S. Donovan, R. Sparks, K. Summers, A. Johnston, and C. Cunningham, "Session Initiation Protocol (SIP) Basic Call Flow Examples," RFC 3665, Jan. 2004. [Online]. Available: https://rfc-editor.org/rfc/rfc3665.txt

[16] E. Schooler, J. Rosenberg, H. Schulzrinne, A. Johnston, G. Camarillo, J. Peterson, R. Sparks, and M. J. Handley, "SIP: Session Initiation Protocol," RFC 3261, Jul. 2002. [Online]. Available: https://rfc-editor.org/rfc/rfc3261.txt

[17] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," *IEEE Transactions on Mobile Computing*, vol. 13, no. 5, pp. 1061–1075, 2014.

[18] S. Uppoor and M. Fiore, "Characterizing pervasive vehicular access to the cellular ran infrastructure: An urban case study," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2603–2614, 2015.

[19] A. M.-C. So and Y. Ye, "On solving coverage problems in a wireless sensor network using voronoi diagrams," in *Proceedings of the First International Conference on Internet and Network Economics*, ser. WINE'05. Berlin, Heidelberg: Springer-Verlag, 2005, p. 584–593. [Online]. Available: https://doi.org/10.1007/11600930_58

[20] T. R. N and R. Gupta, "A survey on machine learning approaches and its techniques:," in *2020 IEEE International Students' Conference on Electrical,Electronics and Computer Science (SCEECS)*, 2020, pp. 1–6.

[21] F. Tang, B. Mao, N. Kato, and G. Gui, "Comprehensive survey on machine learning in vehicular network: Technology, applications and challenges," *IEEE Communications Surveys Tutorials*, 2021.

[22] P. Raut, H. Khandelwal, and G. Vyas, "A comparative study of classification algorithms for link prediction," in *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2020, pp. 479–483.

[23] K. R. Dalal, "Analysing the role of supervised and unsupervised machine learning in iot," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2020, pp. 75–79.

[24] A. Cutler, D. Cutler, and J. Stevens, "Random forests," *Machine Learning - ML*, vol. 45, pp. 157–176, 01 2011.

[25] M.-L. Zhang and Z.-H. Zhou, "A k-nearest neighbor based algorithm for multi-label classification." in *GrC*, X. Hu, Q. Liu, A. Skowron, T. Y. Lin, R. R. Yager, and B. Zhang, Eds. IEEE, 2005, pp. 718–721. [Online]. Available: http://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/grc05.pdf

[26] C. Zhang, J. Shi, L. Li, W. Lin, Y. Wang, L. Gu, Y. Ji, and Z. Feng, "Signaling latency analysis of peer-to-peer sip systems," in *2008 5th IEEE Consumer Communications and Networking Conference*, 2008, pp. 505–509.

[27] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, p. 509–517, Sep. 1975. [Online]. Available: https://doi.org/10.1145/361002.361007

[28] "The Communications Market 2021," Data from Ofcom, UK, July 2021. [Online]. Available: https://www.ofcom.org.uk/research-and-data/multi-sector-research/cmr/cmr-2021

[29] ITU-T, "One-way transmission time," International Telecommunication Union, Geneva, Recommendation G.114, May 2003.