ResearchGate

Why NEMO protocols do not pre-fetch more than one CoA?

Conference Paper · December 2011

DOI: 10.1109/ANTS.2011.6163636

3 authors:

Avik Mitra
The Heritage Academy, Kolkata

Bhaskar Sardar
Jadavpur University

Debashis Saha
Indian Institute of Management Calcutta

Why NEMO Protocols Do Not Pre-fetch More Than One CoA?

Avik Mitra
Department of Information Technology
Jadavpur University
Kolkata, India
avik.mitra2@gmail.com

Bhaskar Sardar
Department of Information Technology
Jadavpur University
Kolkata, India
bhaskargit@yahoo.co.in

Debashis Saha
MIS and Computer Science Group
IIM Calcutta
Kolkata, India
ds@iimcal.ac.in

***Abstract*— Protocols for managing network mobility (NEMO) should reduce handoff latency significantly to provide uninterrupted Internet connectivity. Since the path of a high-speed vehicle is predefined in most cases, it could be beneficial for a NEMO protocol to pre-fetch care-of-addresses of the potential future cells. However, the current NEMO protocols either do not use pre-fetching or use it for at most one cell. While investigating this behavior of the NEMO protocols analytically, we find that pre-fetching beyond the next one cell does not necessarily improve the handoff performance even though the future route of the vehicle could be known beforehand.**

Keywords-NEMO, MIPv6, Handoff, Pre-fetching, Turing machine.

I. INTRODUCTION

The network mobility (NEMO) basic support protocol (NBSP) [1] has been standardized by the IETF to provide seamless Internet connectivity to the passengers of high-speed vehicles, such as bus and train. Each vehicle is equipped with a specialized router, called mobile router (MR), which manages the mobility of the entire in-vehicle network. Since NBSP is an extension of MIPv6 [2], it inherits all the drawbacks of MIPv6 including high handoff latency. According to NBSP, when an MR moves from the coverage area of one access router (AR) to that of another, the MR obtains a care-of-address (CoA) from the new AR (NAR) and registers the future CoA (FCoA) with its home agent (HA). SIP-NEMO [3], SIP-NMG [4] and the protocol by Arita et. al. [5] use this strategy. Obviously, in this strategy the handoff latency is very high. To preempt the acquisition of FCoA, MR may anticipate the next handoff from its predefined route map and predict the NAR. Also, the MR may use link layer (L2) trigger to anticipate handoff, or may log the coordinates of the ARs already visited and use this historical data to predict the next ARs [6]. So the MR can easily pre-fetch the FCoA through the current AR (CAR), and register the FCoA with the HA. The handoff latency in this strategy is much lower than the previous strategy. The protocol proposed by Mussabir et. al. [7] and HiMIP-NEMO [8] use L2 trigger to anticipate handoff. As reported in [7] and [12], advanced registration at internet layer (L3) of TCP/IP may also reduce the delay in L2 handoff and delay in receiving the L3 advertisement from AR by a substantial amount (~157ms [12]). Simultaneously, the signaling overhead may be reduced by as much as 50% [7].

From the above discussions, it is clear that the number of FCoAs (denoted by b in this work) in the first strategy is 0 and in the second strategy is 1. However, since the ARs to be followed by an MR could be known in advance, we can utilize this knowledge and design a NEMO protocol that can pre-fetch any number of FCoAs to be used in the future cells. So the value of b should be ideally greater than 1 and can attain a maximum value, say n, equal to the total number ARs to be visited during the whole journey. Thus, we can categorize the NEMO protocols in terms of the variable b. In Figure 1, we have mapped the existing NEMO protocols along the "b-line". We observe that the currently available protocols are mapped either at b=0 or b=1, thereby setting the current upper limit for b to 1. Naturally the question arises why there be no protocol with b>1. Certainly, the answer lies in the knowledge about the dependency of handoff latency on b. In this paper, we investigate whether there is any gain by allowing b to attain a value larger than 1. To the best of our knowledge, till now, this issue has not been addressed by the NEMO research community. First, we prove that the problem of determining whether pre-fetching two or more FCoAs is right or wrong is yet undecidable, and then we investigate the effect of over pre-fetching on the protocol performance.

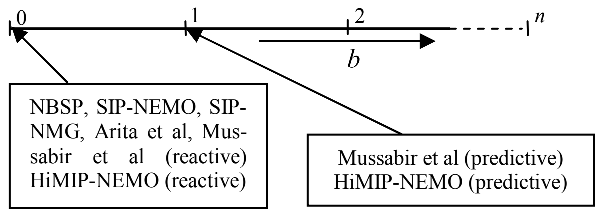

Figure 1: Mapping of NEMO protocols on b-line

The rest of the paper is organized as follows. Section II describes the network model, Section III contains the analysis on FCoA pre-fetching, and Section IV concludes the paper.

II. NETWORK MODEL

We consider a model of NEMO (as shown in Figure 2) having the following characteristics. The area covered by an AR (called a cell) is hexagonal with the AR placed at the center of the cell. The cells are of equal size with circumradius r.

A. Definitions (refer to Figure 2)

1. Tier-0 cell: A cell at which the MR resides at the start of observation.
2. Tier-1 cell: All the neighbors[1] of tier-0 cell.
3. Tier-k cell (k>1): Those neighbors of tier-(k-1) cells, which do not belong to tier-h (0≤h<k-1).
4. Tier-i cell is a higher tier cell than a tier-j cell if i >j.

---

[1] *Cell i is neighbor of cell j if the distance between their centers is* $\sqrt{3}\,r$*, i.e., cell i is adjacent to cell j.*

B. Assumptions on the behavior of MR

1. Initially, the MR is in cell 0 (Figure 2).
2. From a tier-k cell, the MR does not move to either tier-(k-1) cells or tier-k cells but moves only to one of the neighboring tier-(k+1) cells.
3. Probability that the MR, residing in a tier-k cell, moves to any of the tier-(k+1) cells is same for all tier-(k+1) neighbors.
4. For prediction, an algorithm W is associated with the NEMO protocol installed in the MR. The MR uses W to predict next higher tier cell[2] and then pre-fetches the FCoA using a suitable mechanism.

From Figure 2, we see that from cell-0 the MR can move to any cell between cell-1’ to cell-6’. From cell-1’, it can move either to cell-1’’ or cell-12’’ or cell-2’’. So, a path traversing cell-0, cell-6’, cell-12’’, and cell 17’’’ is a valid path whereas a path traversing cell-0, cell-6’, cell-1’, cell-12’’, and cell-17’’’ is an invalid path.

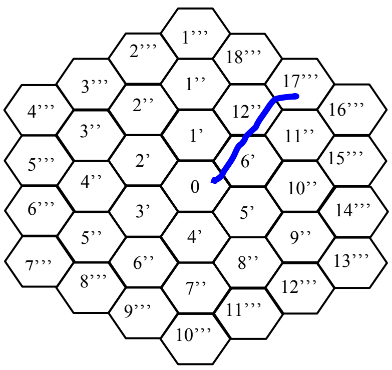

Figure 2: Network model for NEMO. Cell-0 is tier-0 cell, cell-1’ to cell-6’ are tier-1 cells; cell-1’’ to cell-12’’ are tier-2 cells; cell-1’’’ to cell-18’’’ are tier-3 cells.

III. Analysis of the Effect of Pre-fetching More than One FCoA

Let us assume that MR logs the sequence of its positions (i.e., two-dimensional coordinates (x, y) of ARs visited) in a list, where (x, y) are integers. Since a set of integers is countable, each coordinates can be replaced by coordinates $(x^{/}, y^{/})$ where $x^{/}$, $y^{/}$ ε N (set of natural numbers). A path of length n, followed by the MR, is then a sequence of coordinates, Z = $\{(x_i^{/}, y_i^{/}): 1 \le i \le n\}$. Now, we represent coordinates using unary coding [10]. Each $x^{/}$ (or $y^{/}$) being natural numbers, the decimal representation of $x^{/}$ (or $y^{/}$) consists of digits from 0 to 9. In unary encoding, 0 is encoded as 1, 1 as 11, 2 as 111 and so on. We use 0 to separate digits, i.e., we encode 23 as 11101111. We separate coordinate components by 00, and coordinates by 000. Thus, we encode a path Z = {(1, 2), (14, 10)} as 110011100011011111001101.

The prediction algorithm, W, (assumption 4 of section II.B) takes as input the description of a path (encoded as described above) and tries to predict the next coordinate in the sequence. We say that the output from W is right or wrong, iff we know the actual output for an event corresponding to which W is predicting. Though a couple of algorithms are available in the literature [13][14] to predict the mobility of mobile devices, they rely heavily on certain assumptions, and none of them guarantees 100% success. This motivates us to ask the imminent question: whether the success rate of mobility prediction can be 100%; if not, then why, and more importantly, whether it can be decided at all by a Turing machine. Since, in general, the output for a particular event cannot be decided by a Turing machine [10], we cannot say whether the prediction is correct. We consider Turing machine for the sake of generalization because the other similar models of computation can be reduced to Turing machine.

MR uses NBSP which is nothing but a sequence of well defined algorithms. Since an algorithm can be represented by a Turing machine, MR can be represented by a Turing machine. Also, the MR employs W, the prediction algorithm, which itself is a Turing machine. So MR is a Turing machine which uses another Turing machine W. We further assume that an event can be represented by a Turing machine, E, and vice versa [11]. Let us first assume that there exists an algorithm A that takes as input the description of an E and input to the event, string w, and decides whether an event will produce an output at all. This is same as finding a Turing machine, T (which implements A), that will terminate in a final state if an event produces an output, else terminate in a non-final state if the event does not produce an output [10]. The algorithm A says ‘yes’ if E produces an output and it says ‘no’ otherwise. If A says ‘yes’, then we compare the outputs from E and W. If the outputs match, then the MR concludes that the prediction is correct and it pre-fetches FCoAs for higher tier cell. However, if the outputs do not match or A says ‘no’, then the MR concludes that the prediction is wrong and does not pre-fetch any FCoA for the higher tier cell.

We follow the definition as given in [10] which states that the Turing machine, T, when given an input x, produces an output v if $q_0 x \vdash^*_T q_f v$, where q_f is a final state and q_0 is the initial state. ‘v’ is produced on the tape of T. Using the algorithm A we create another algorithm B that takes as input the description of a Turing machine, M, a string w and decides whether M accepts w or not by invoking A on M and w. The Turing machine M accepts w if the algorithm B outputs ‘yes’. Similarly, the Turing machine M does not accepts w if the algorithm B outputs ‘no’. Thus, B is a membership algorithm for recursively enumerable languages [10].

It is clear that if the algorithm A terminates then the algorithm B will terminate. On the other side, if the algorithm A does not terminate then the algorithm B will not terminate. Hence, the membership problem for recursively enumerable languages is *reduced* to the problem of whether there will be an output from an event, E. But existence of B is a contradiction [10] because no membership algorithm exists for recursively enumerable languages. Hence, the algorithm B does not exist; consequently, the algorithm A also does not exists i.e., in general, the problem of deciding output from an event is undecidable.

Since, in general, we cannot decide programmatically what would be an output from an event, we cannot decide deterministically whether the prediction from W is right or wrong. We

---

[2] In broad-gauge railroads, the radius of curvature is not less than 175 meters [9], i.e., there is no sharp turn along the tracks.

have not assumed anything regarding the internal working of W. Moreover, the MR logs its sequence of its position (or ARs till covered) without regard to cell geometry. Hence, the undecidability problem holds irrespective of prediction algorithm or cell-geometry. So, in the following, we analyze the success rate of prediction and correctness of pre-fetching more than one future cell, using probabilistic technique.

We consider that initially the MR is in tier-0 cell. So the probability that the MR moves into any one of the tier-1 cells is (1/6). We assume that algorithm W predicts cell-1' as possible tier-1 cell. So the probability that the prediction is correct is:

$$P(cell-1') = \frac{1}{6} \quad (1)$$

Now, the algorithm W is invoked again, and let us assume that it predicts cell-12'', a tier-2 cell, as the next cell (thus, value of b becomes 2). The probability that the path of the MR will traverse cell-1' and cell-12'' is:

$$\begin{aligned} &P(cell-1' \cap cell-12'') \\ &= P(cell-1') \times P(cell-12''/cell-1') \end{aligned} \quad (2)$$

According to assumption 3 of section IIB we have: $P(cell-1') = \frac{1}{6}$ and $P(cell-12''/cell-1') = \frac{1}{3}$

So, from Equation (2), the probability that the predicted traversal is correct is:

$$P(cell-1' \cap cell-12'') = \frac{1}{18} \quad (3)$$

In Equation (3), the probability of correct prediction is decreased. Now, we generalize the above case. When the MR is in cell-0, we assume that W predicts A_1 (a tier-1 cell), A_2 (a tier-2 cell),…,A_n (a tier-n cell). Then, the probability that this prediction is correct is given as:

$$\begin{aligned} &P(A_1 \cap A_2 \cap \ldots \cap A_n) \\ &= P(A_1)P(A_2/A_1)P(A3/A_1 \cap A_2)\ldots P(A_n / \bigcap_1^{n-1} A_i) \end{aligned} \quad (4)$$

Also, we assume that W predicts A_{n+1} as next tier-(n+1) cell. The probability that this prediction is correct is

$$\begin{aligned} &P(A_1 \cap A_2 \cap \ldots \cap A_n \cap A_{n+1}) \\ &= P(A_1 \cap A_2 \cap \ldots \cap A_n) \times P(A_{n+1} / \bigcap_1^{n} A_i) \end{aligned} \quad (5)$$

By carefully observing Figure 2, we find that every tier-k cell (k≥2) has either two tier-(k+1) neighbors or three tier-(k+1) neighbors. Hence, we have:

$$\begin{aligned} P\left(A_{n+1} / \bigcap_1^{n} A_i\right) &= \frac{1}{2}, \text{if } A_n \text{ has 2 tier - (n + 1) neighbors} \\ &= \frac{1}{3}, \text{if } A_n \text{ has 3 tier - (n + 1) neighbors} \end{aligned} \quad (6)$$

So, from Equation (5) and (6) we can write:

$$P(A_1 \cap A_2 \cap \ldots \cap A_n \cap A_{n+1}) < P(A_1 \cap A_2 \cap \ldots \cap A_n)$$

Applying Equations (6) and (5) to (4) successively, we get by induction:

$$P(A_1) > P(A_1 \cap A_2) > \ldots > P(\bigcap_1^{n} A_i) > P(\bigcap_1^{n+1} A_i) \quad (7)$$

So the more we predict about the future traversal (b>1), the less is the probability of correct prediction. From Equations (5), (3), and (6), we can conclude that the probability of getting correct prediction decreases by (1/2)×100% = 50% or (1-(1/3))×100% = 66.7% (approx.) for every tier equal to or higher than 1. So, the best we can do is to restrict W to predict only about tier-1 cells (i.e., setting b=1), for which the success rate is about 100×(1/6)%=16.7% and allow W to predict about tier-2 cells when the MR has actually moved to a tier-1 cell. This observation is in agreement with the current NEMO protocols which do not pre-fetch FCoA beyond immediate NAR.

IV. CONCLUSIONS

The handoff performance of NEMO protocols cannot be improved much by pre-fetching CoAs for the future cells. We have investigated this issue and found that pre-fetching beyond the immediate NAR does not necessarily improve the handoff performance. So we justify anticipating only the immediate NAR (as is done in most protocols) even if the path to be followed by the high-speed vehicle is predefined.

REFERENCES

[1] Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network mobility (NEMO) basic support protocol", RFC 3963, January 2005.

[2] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[3] C. M. Huang, C. H. Lee, and J. R. Zheng, "A novel SIP-based route optimization for nerwork mobility", IEEE JSAC, vol. 24, issue 9, pp. 1682-1691, september 2006.

[4] Y. C. Tseng, J. J. Chen, and Y..L. Cheng, "Design and implementation of SIP-based mobile and vehicular wireless network with push mechanism", IEEE TVT, vol. 56, no. 6, pp. 3408-3420, November 2007.

[5] T. Arita and F. Teraoka, "A fast handoff mechanism using cross-layer collaboration for mobile networks in high-speed trains", Asia future internet forum, Winter school, February 2010.

[6] C. Cheng, R. Jain, and E. V. D. Berg, "Location prediction algorithms for mobile wireless systems", Wireless Internet Handbook: Technologies, Standards & Applications, CRC Press, pp. 245-263, 2003.

[7] B. Mussabbir, W. Yao, Z. Niu, and X. Fu, "Optimized FMIPv6 using IEEE 802.21 MIH services in vehicular networks", IEEE TVT, vol 56, no. 6, pp 3397-3407, November 2007.

[8] C. W. Lee, M. C. Chen, and Y. S. Sun, "A network mobility management scheme for fast QoS handover", WICON, November 2008.

[9] Railway curves, IRICEN, 1st edition, January 2010.

[10] P. Linz, "An introduction to formal languages and automata", 3rd edition, Narosa publishing house, 2004.

[11] W. Baer, "Using the turing machine to simulate physics", 2009, on-line: http://faculty.nps.edu/baer/CompModphys/cem_journal_turingToSim/cem_journal_turingToSim.htm

[12] G. Xie, J. Chen, H.Zheng, J. Yang and Y.Zhang, "Handover latency of MIPv6 Implementation in Linux", IEEE GLOBECOM 2007.

[13] G.Y. Liu, G.Q. Maguine Jr., "A Predictive Mobility Management Algorithm for Wireless Mobile Computing and Communications", IEEE ICUPC 1995.

[14] R. Sanakar, N. Savkoor, "A Combined Prediction System for Handoffs In Overlaid Wireless Networks", IEEE ICC 1999.