0-1 Knapsack Problem Approach for Multicast Agent in NEMO System

Rashid Saeed[1], Abdelmuizz Eisa[2], Raed Alsaqour[3*]

[1]Faculty of Engineering, Sudan University of Science and Technology (SUST), Khartoum, Sudan

[2]Faculty of Engineering and Technology, University of Gezira, Khartoum, Sudan

[3]School of Computer Science, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Selangor, Malaysia

{eng_rashid@ieee.org, mo22ap@gmail.com, raed@ftsm.ukm.my}

Abstract— Multicast provides an effective group communication mechanism in IP layer to reduce the redundant packets transmission compared with the unicast. The knapsack problem is a problem in combinatorial optimization when we have given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible. This paper discusses 0-1 Knapsack approach to use by NEMO's multicast agent (MA) to ensure the forwarding data to their destinations in combinational optimize selection manner, and acceptable throughput.

Keyword-Multicast Agent, Mobile Router, Remote Subscription, Tunneling, Knapsack Problem

I. INTRODUCTION

In mobile networks (NEMO), the mobility function moved from mobile nodes to a mobile network's router (MR). That router is able to change its attachment point to the Internet in a manner that is transparent to attached node [1]. And mobile multicast becomes more important because it can provide many kinds of real time applications.

The MR forwards the multicast data via a dedicated tunnel to the multicast agent (MA) which to choose the MR has decided on. In case of subscription, as receiver, the request is forwarded utilizing the MLDv2 [2-5] protocol. In case of transmission of originating traffic addressed to a multicast group, the traffic data are sent to the MA for further distribution. Corresponding multicast traffic received at the MA is subsequently forwarded by the MA back to the MR via the bi-directional tunnel. Detection of a set of candidate MAs should to be performed by the MR before applying an algorithm to decide on the best suited serving MA for a specific multicast service. The set may contain the (MR's Home Agent, the Current "old" or future "new" Access Router AR) (AR/oAR/nAR) [6, 7]. So MA is a multicast router enabled to receive and forward multicast traffic by a multicast routing protocol. All multicast data aimed at nodes outside the NEMO are tunneled to the MA to separate mobility issues from the multicast tree.

Multicast traffic is characterized by session duration, direction to or from the NEMO and service parameters like QoS requirements. This information must be transferred to the MR together with the subscription request. A potential MA is characterized by existing subscription status, path distance to the MR and grad of service (in term of throughput or total number of groups served by MA).

The knapsack problem (KP) is a problem in combinatorial optimization when we have given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible [8]. See Fig. 1

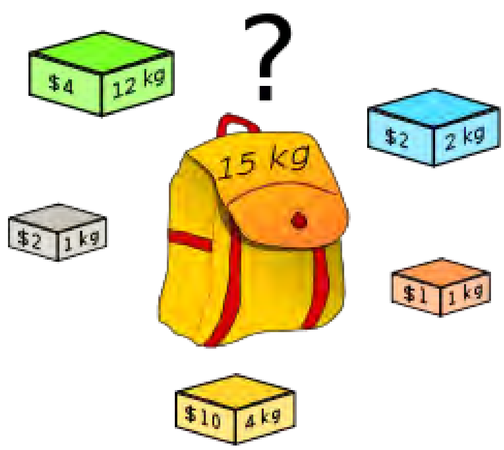

Fig. 1. Knapsack Scenario

The problem is how to select maximum numbers of k^{th} items, from x_k copies, has v_k values and w_k weights.

i.e.

$$Max \quad \sum_{k=1}^{n} v_k \, x_k \tag{1}$$

Subject to:

$$\sum_{k=1}^{n} w_k \, x_k \leq W \tag{2}$$

where W is the maximum size of Knapsack.

In equation 1;

$if \quad x_k \in \{0,1\} \; (0-1 \; Knapsack \; problem)$

$if \quad x_k \in \{0,1,..,C_n\} \; (Bounded \; Knapsack \; problem)$

$if \quad x_k \in \{0,1, \dots \dots ..\} \; (Unbounded \; Knapsack \; problem)$

This paper discusses the 0-1 Knapsack approach that could be supported to NEMO's MA to ensure the forwarding data to their destinations in combinational optimize selection manner and acceptable throughput.

The rest of the paper is organized as follows. Section 2 presents the network mobility multicasting; section 3 discusses the multicast agent's paths and selection them according 0-1 knapsack concept. Section 4 presents a ways to solve 0-1 KP. Section 5 shows the performance results for ideas mentioned in section 3 and section 4, conclusion and references under section 6 and section 7 respectively.

II. NEMO MULTICASTING

Multicast provides an effective group communication mechanism in IP layer to reduce the redundant packets transmission compared with the unicast. Fixed multicast is based on host group model [9] which consists of multicast route protocol and multicast group control protocol. Multicast routing protocols constructs the multicast delivery tree and maintains the multicast states in multicast routers, while multicast control protocol is used by multicast router and host to exchange the multicast membership information [10, 11].

Traditional multicast routing protocols have been designed for static network topology and the multicast delivery tree is stable or little changed. For mobile multicast, these assumptions changed. Directly applying traditional multicast route protocols to NEMO scenario may result in multicast session disruption. To improve the performance of multicast services, it has to solve the following general problems. The foundational problem in mobile multicast is the reconstruction of multicast delivery tree, The problems related to multicast packets transmission and reducing leave/rejoin receivers delays can improve the multicast throughput and services.

For multicast NEMO packets, two basic methods have been proposed by Internet Engineering Task Force (IETF) bidirectional tunneling (BT) and remote subscriptions (RS) methods [12]. BT is transparent and has lower join latency, but introduces triangular routing and additional transmission overheads. While RS has effective routing, but increases long join delay especially when foreign networks do not have the multicast states of mobile network interested, and even result in multicast service disruption if foreign networks do not support multicast. Later IETF proposed to use agent base (AB) method to support multicasts in NEMO because it is combine between BT and RS advantages and prevent they most drawbacks.

As shown in Fig. 2 the multicast group members are send and receive a multicast traffic through group of forwarders that could be containing a combinations from home agent, top level MR access router, multicast agent and new/old access router (HA, TLHA, AR/MA, old-AR, new-AR). While the multicast group members are two categories, one is local mobile nods (LMN) and other is visited mobile nodes (VMN). The VMNs could be mobile nodes or other MR has NEMO cluster try to jointing/leaving to the multicast group.

The low and high speed of NEMO is affect in the NEMO system performance when we apply the BT or RS techniques by tunneling amplification or long delay due to the high rate of subscription respectively or combined them in agent base approach. In agent based procedure the TLMR establish new functions like: Forwarding the multicast group management messages, keeping and update address and deciding on optimum multicast agent based on movement [13].

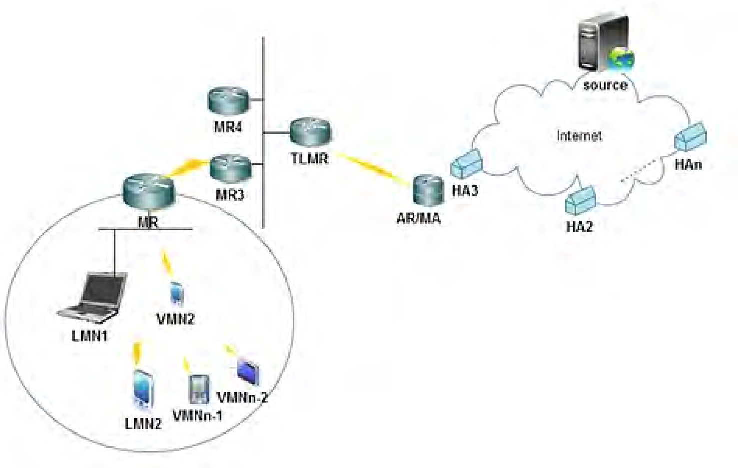

Fig. 2. Nested Topology of NEMO Scenario

Instead of performing selection and decision on the site of the multicast agent at the AR we propose to adding 0-1 Knapsack algorithm to improve the performance of NEMO multicast system.

A. Paths Selection

A path selection mechanism is required to select among the multiple available paths. Depending on the NEMO multihoming configuration involved, the differences between the paths may affect only the part between the HA and the MR, or they may affect the full end-to-end path. In addition, depending on the configuration, path selection may be performed by the HA(s), the MR(s) or the hosts themselves through address selection.

Also, the multiple paths available may differ in more than just the tunnel between the MR and the HA, since the usage of different prefixes may result in using different providers, hence in completely different paths between the involved endpoints. In this case, additional dynamic path selection mechanisms for the end-to-end path selection may be needed. This mechanism may be closely related to source address selection mechanisms within the hosts, since selecting a given address implies selecting a given prefix, which is associated with a given ISP serving one of the home networks. That's paths could be selected by:

HA (it should be able to select the path based on some information recorded in the binding cache), MR (it should be able to select the path based on router advertisements), The MNN (it should be able to select the path based on default router selection [14]) and may be selected by the user or the application: e.g. in case where a user wants to select a particular access technology among the available technologies for reasons like cost or data rate.

Additional case for selection may depend a combination of any of the above: a hybrid mechanism should be also available, e.g. one in which the HA, the MR, and/or the MNNs are coordinated to select the path in combination manner.

III. MULTICAST AGENT TREE FORMULATION

Suppose the MA multicasting a single data item through single path; i.e. $\{p, \mathbb{S}, \tau\}$: means deliver data to a set $\mathbb{S}$ of mobile network nodes (MNNs); how to choose a shortest path to achieve the delivery ratio p within the time constraint τ. In general case: $\{p, \mathbb{S}_1 \dots, \mathbb{S}_n, z_1, \dots, z_n, \tau\}$ means deliver data to a set $d_1, \dots, d_n$ with size $z_1, \dots, z_n$, from data source to destination sets $\mathbb{S}_1 \dots, \mathbb{S}_n$; how to choose a shortest paths to achieve the delivery ratio p within the time constraint τ [15].

For MA, we need to maximization the objective function (amount of data sets delivered to their destinations ($v_k x_k$)) subject to routing constrains (i.e. $x_k w_k$).These constrains should satisfy the performance requirements [8, 16].

$$max \sum_{k=1}^{n} x_k\, v_k$$

$$\text{s.t.} \quad \sum_{k=1}^{n} w_k\, x_k \le W \tag{3}$$

where $x_k \in \{0,1\}$ indicates whether of routes selected by MA and the constraint indicate that selection should satisfy the performance requirements in delivery ratio, and delay [7].

For example the multicast selection method as 0-1 Knapsack problem (KP) of three items with its values, are shown in Fig. 3.

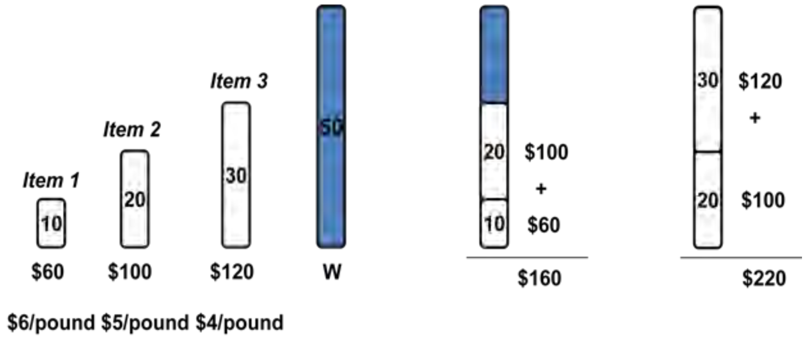

Fig. 3. 0-1 KP

The target is to maximize the value of a knapsack that can hold at most *W* units from a list of items. So 0-1 KP means you either take an item or not.

On other hand the KP is a problem in combinatorial optimization: Given a set of items, each with a weight and a value, determine the number of each item to include in a collection. The total weight is less than or equal to a given limit and the total value is as large as possible. The most common problem being solved is the 0-1 KP, which restricts the number x_k of copies of each kind of item to zero or one.

IV. Solving 0-1 Knapsack Problem

Several algorithms are freely available to solve KPs, based on dynamic programming approach [17], branch and bound approach [8, 13] or hybridizations of both approaches [16]. The dynamic programming approach is selected to solve 0-1 KP because it is more efficient than branch and bound methods and it takes less time than other methods.

The dynamic programming is a method for solving complex problems by breaking them down into simpler sub-problems. It is applicable to problems exhibiting the properties of overlapping sub-problems and optimal substructure .

A. Dynamic-Programming Solution to the 0-1 KP

For showing the solution mechanism, let us suppose the algorithm takes as input the maximum weight *W*, the number of items *n*, and the two sequences $v = (v_1, v_2, . . . , v_n)$ (sequences of values) and $w = (w_1, w_2, . . . , w_n)$ (sequences of waits). It stores the *C[i, j]* values in the table, that is, a two dimensional array, *C[0 . . n, 0 . . w]* whose entries are computed in a row-major order. The first row of *C* is filled in from left to right, then the second row, and so on. At the end of the computation, *C[n, w]* contains the maximum value that can be picked into the knapsack.

Dynamic 0-1-knapsack (v, w, n, W)

```
FOR w = 0 to W
     DO C[0, w] = 0
    FOR i=1 to n
        DO C[i, 0] = 0
       FOR w=1 TO W
           DO IF  w_i ≤ w
               THEN v_i + C[i-1, w-w_i]
                      C[i, w] = v_i + C[i-1, w-w_i]
               ELSE C[i, w] = C[i-1, w]
C[i, w] = C[i-1, w]
```

B. Running Time of 0-1 KP Solution

If we used branch and bound algorithm (straightforward algorithm) to solve 0-1 KP, we have *n* items, there are 2^n possible combinations of items. We go through all combinations and find the one with maximum value and with total weight less or equal to *W*. so the running time will be $O(2^n)$.

But in selected solution based on a dynamic programming algorithm, the recursive formula for sub-problems been to compute *V[k,w]*, i.e., to find an optimal solution for x_k = {items labeled 1, 2, .. k} in a knapsack of size *W* [8].

```
for w = 0 to W                                          O(W)
        B[0,w] = 0                                      Repeat    n
for i = 1 to n                                          times
        B[i,0] = 0                                      O(W)
for i = 1 to n
    for w = 0 to W
        if wi<= w "item i can be part of the solution"
            if bi+ B[i-1,w-w] > B[i-1,w]
                    B[i,w] = bi+ B[i-1,w- w]
          else
                    B[i,w] = B[i-1,w]
        else B[i,w] = B[i-1,w]  "wi> w"
1
```

i.e. the running time of this algorithm is O(nW). So the dynamic programming approach is better than straightforward algorithm.

V. PERFORMANCE EVALUATION

We can suppose the maximum routing capacity of MA presented by the total Knapsack size and the total number of items (number of multicast paths) should fill in the knapsack (should be routed by MA) is ten.

1. The selected items represented by selected multicast paths.
2. The maximum KP represented by maximum routing capacity of MA.
3. The total number of items represented by the number of multicast paths.
4. The weight of item represented by amount of multicast data per path.

Fig. 4 shows the paths selected by MA (sel. values) when the maximum number of multicast paths should be routed is 10, found nine of them been routed according to data amount (weight) per path in combinational optimization selection manner. Mean that the total amount of data must route on 10 paths is 5000 data units, found that 4612 data units been routed. Figs. 5 and 6 shows the selection effect of MA if the total amount of data per total available paths is decreasing.

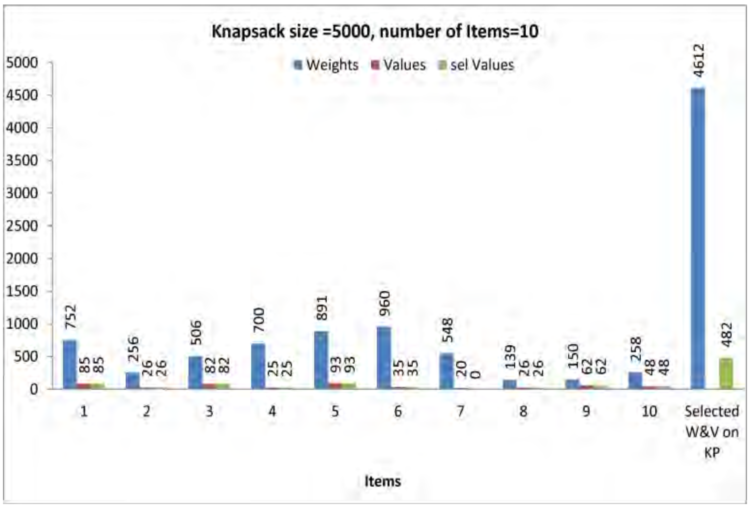

Fig. 4. KP size =5000 & number of items=10

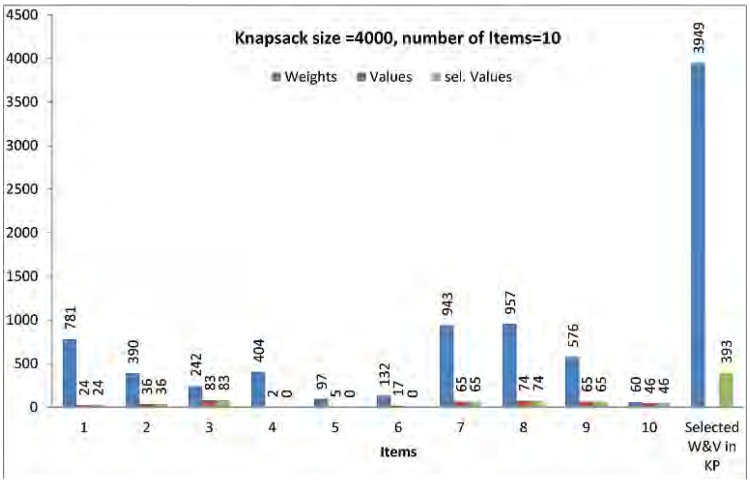

Fig. 5. KP size =4000 & number of Items=10

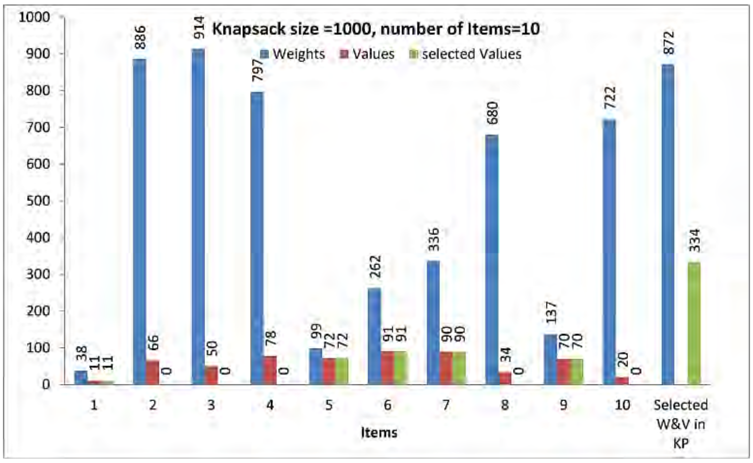

Fig. 6. KP size =1000 & number of Items=10

VI. Conclusion

The multicast agent schemes increase efficiency of multicast traffic in mobile networks and it adds new entity to the network and more addressing complexity. The Knapsack algorithm gives us a new idea to improve the performance of AB multicasting in NEMO systems.

In this paper, the selection depends to the main factor *v (value)*. It lets unspecified to ensure the door open to different constraints of selections in future. Means *v* could represent any factor like, delay in routing condition in critical cases of considerations, critical handover or dedicated tunnels when the cluster in high speed.

Acknowledgment

This study was partially funded by the University Kebangsaan Malaysia under Grant Nos. UKM-GUP-2012-089 and FRGS/1/2012/SG05/UKM/02/7.

References

[1] M. Kim, T.-J. Lee, and H. Choo, "NeMRI-based multicasting in network mobility," in Computational Science and Its Applications-ICCSA 2006, ed: Springer, 2006, pp. 710-719.

[2] H. Holbrook, B. Cain, and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast," Request for Comments, vol. 4604, 2006.

[3] S. Deering, W. Fenner, and B. Haberman, "Multicast listener discovery (MLD) for IPv6," RFC 2710, October 1999.

[4] R. Vida, L. Costa, S. Fdida, S. Deering, B. Fenner, I. Kouvelas, et al., "Multicast listener discovery version 2 (MLDv2) for IPv6," RFC 3810, June 2004.

[5] B. Fenner, H. He, B. Haberman, and H. Sandick, "Internet Group Management Protocol (IGMP)/Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")," RFC 4605, vol. 4605, 2006.

[6] H. Dirk, "Agent-based multicast support for moving networks (NEMO) [Online]. Available: http://tools.ietf.org/html/draft-von-hugo-multimob-agents-03," ed, 2009.

[7] R. A. Saeed and A. A. A. Eisa, "Multicasting in network mobility using multicast anchor agent," in Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on, 2013, pp. 423-428.

[8] S. Martello, D. Pisinger, and P. Toth, "Dynamic programming and strong bounds for the 0-1 knapsack problem," Management Science, vol. 45, pp. 414-424, 1999.

[9] S. E. Deering, "Host extensions for IP multicasting [Online]. Available: http://tools.ietf.org/html/rfc1054," 1988.

[10] T. Abdullah, R. Alsaqour, M. Abdelhaq, R. A. Saeed, and O. Alsukour, "Reliable Multicast Error Recovery Algorithm in Ad-Hoc Networks," Research Journal of Applied Sciences, Engineering and Technology, vol. 4, pp. 4656-4662, 2012.
[11] T. Alahdal, R. Alsaqour, M. Abdelhaq, R. Saeed, and O. Alsaqour, "Reliable Buffering Management Algorithm Support for Multicast Protocol in Mobile Ad-hoc Networks," Journal of Communications, vol. 8, pp. 136-150, 2013.
[12] K. Park, S. Han, K. Kim, and J. Hong, "Dynamic Multicast Tree Construction for NEMO," draft-park-nemo-dyn-multicast-tree-00, Expired, 2005.
[13] S. Martello and P. Toth, Knapsack problems: algorithms and computer implementations: John Wiley & Sons, Inc., 1990.
[14] C. Ng, E. Paik, T. Ernst, and M. Bagnulo, "Analysis of multihoming in network mobility support," Work in Progress, 2007.
[15] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing, 2009, pp. 299-308.
[16] D. Pisinger, "Algorithms for knapsack problems," PhD Thesis, University of Copenhagen, Denmark, 1995.
[17] R. Andonov, V. Poirriez, and S. Rajopadhye, "Unbounded knapsack problem: Dynamic programming revisited," European Journal of Operational Research, vol. 123, pp. 394-407, 2000.