Supplemental Material for "Distributed Placement of Autonomic Internet Services"

Panagiotis Pantazopoulos, Merkouris Karaliopoulos, *Member, IEEE,* and Ioannis Stavrakakis, *Fellow, IEEE*

To offload our manuscript we have placed a portion of our work here. In Section 1 we derive analytical closed-form expressions for the $(w)CBC$ metric variation in regular topologies while next, we draw insights regarding the cDSMA performance from the degree distributions of the studied ISP topologies. Section 3 presents the way the cDSMA practical implementation operates under multipath routing (MP). Section 4 experimentally investigates whether overload phenomena can occur with the cDSMA implementations. Finally, we discuss a wide range of solutions that fall in the relevant data replication and placement category.

1 Metric computation for regular network topologies

Closed-form expressions for $wCBC$ are not easy to obtain except for scenarios with uniform demand and regular topologies. The following two Propositions provide the closed-form expressions for CBC, *i.e.*, $wCBC$ for $w_n = 1, \forall n \in V$, in two instances of regular network topologies, the ring and the two-dimensional (2D) grid.

Proposition 1.1: In a ring network of N nodes, the CBC value of a node u with respect to another node t are given by:

$$CBC_{ring(N)}(u;t) = \begin{cases} \lceil \frac{N-1}{2} - d(u,t)) \rceil^+ & N = 2k \\ \lceil \frac{N+1}{2} - d(u,t)) \rceil^+ & N = 2k+1, \end{cases}$$

where $k \in \mathbb{Z}^+$, $\lceil x \rceil^+ = max(x,0)$ and $d(u,t)$ is the minimum hop count distance between nodes u and t along the ring.

Proof: The proof is straightforward. There is one minimum hop count path between all pairs of nodes in the ring. The only exception concerns nodes $N/2$ positions away the one from another in rings with even number of nodes, where there are two shortest paths. For given destination node t, the $CBC(u,t)$ value is only increased by those shortest paths that encompass the intermediate node u. Due to the ring symmetry, their number only depends on the distances between nodes u and t and decreases by one for each additional hop away from t. Summing them over the respective half of the ring, yields the result. □

Proposition 1.2: Consider a MxN rectangular grid network, where nodes are indexed inline with their position in the grid, *i.e.*, node (i,j) is the node located at the i^{th} row and j^{th} column of the grid. The CBC value of node u at position (a,b) with respect to node t at position (k,l) is given by (1).

Proof: For the 2D grid, the problem degenerates into the enumeration of shortest paths between two grid nodes [1]. The denominator of (1) expresses the number of shortest paths between two arbitrary nodes $(row, column)$ coordinates (i,j) and (k,l), whereas the numerator of (1) equals the number of those paths going through a node with coordinates (a,b). We then sum the ratios over all grid nodes with shortest paths to node $t = (k,l)$ encompassing node $u = (a,b)$. □

2 Degree distribution of the real-world topologies

For the evaluation of the theoretical cDSMA algorithm as well as its practical implementations we have employed a set of real-world ISP snapshots. To gain further insights on our results we present the degree distribution of a subset of these topologies in fig 3.

The general structural characteristics of the considered ISP topologies differ from the synthetic topologies in section 6.1 of the main paper; they exhibit neither the regularity of grids nor the extreme degree variance and hub nodes of B-A like topologies. However, the presence of a few high degree nodes in almost all considered topologies, ends up trapping the migrating service in a way that was also evidenced in B-A graphs. More specifically, the service trapping incidents in the B-A graphs resulted from the combination of high degree nodes with small average paths. Namely, when we increase the α percentage of nodes within the 1-median subgraph, we only include in the G_{Host} those nodes that are immediate neighbors to the current hub node serving as host. The G_{Host} spans around the hub node, which remains the lowest-cost location within the subgraph so that the service migration process is terminated. On the other hand, the real-world topologies are characterized by greater average shortest paths, yet may exhibit similar

• *The authors are affiliated with the Department of Informatics and Telecommunications, National & Kapodistrian University of Athens, Ilissia, 157 84 Athens, Greece.*
E-mail: {ppantaz, mkaralio, ioannis}@di.uoa.gr

$$CBC_{grid(M,N)}(u;t) = \sum_{i=1}^{M}\sum_{j=1}^{N} \frac{\binom{|b-j|+|a-i|}{|a-i|}\binom{|l-b|+|k-a|}{|k-a|}}{\binom{|l-j|+|k-i|}{|k-i|}} \mathbb{1}_{\{|l-j|+|k-i|==|b-j|+|a-i|+|l-b|+|k-a|\}} \quad (1)$$

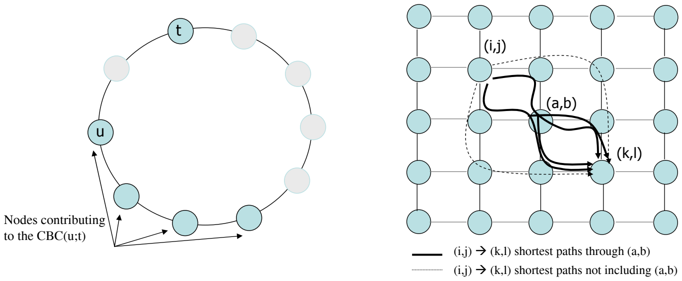


Fig. 1. Conditional Betweenness Centrality in regular topologies.

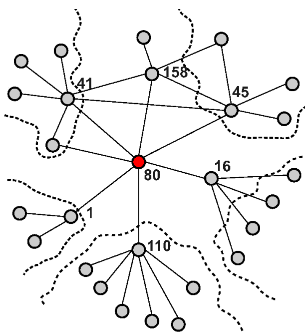


Fig. 2. Partial snapshot of Dataset 33. The hub node 80 constitutes a trap for cDSMA especially under uniform demand; a service whenever generated around that node, ends up at 80 which is a local minimum yielding similar trap phenomena to those observed in B-A graphs.

phenomena. A relevant trap we have identified in Dataset 33 involves exactly a hub node, whose first neighbors are not fully linked with each other (see fig. 2). The service is often trapped there and the cDSMA requires far more G_{Host} nodes to achieve near-optimal placements than it does for the rest of the ISP topologies.

3 cDSMA PRACTICAL IMPLEMENTATION UNDER MULTI-PATH (MP) ROUTING

We discuss the cDSMA practical implementation in the same step-by-step fashion as did with the single-path (SP) routing case. Details are provided only for the points where the multipath (MP) routing option differentiates the cDSMA implementation.

3.1 Service host advertisement

This step is carried out in the same way regardless the employed routing protocol.

3.2 Reporting of local wCBC estimates and inference of the 1-median subgraph

As each measurement-reporting message travels on its shortest path towards the $Host$, it records all nodes lying on it. The G_{Host} subgraph as inferred by the current host node exhibits attributes that depend on the employed routing protocol.

Under multipath routing data packets make use of more than one shortest routes towards a single destination, effectively balancing the traffic load across these paths. The resulting 1-median subgraph is not a connected tree as under SP routing, and the distance of any G_{Host} node from the $Host$ is now upper-bounded only by the network diameter. For example, in fig. 4.c the demand traffic that flows as a whole through the 1-median subgraph node M is subsequently split across three different paths leading to the current service host node A. If nodes L, P and N exhibit low native demand values, they may well not be selected by the $wC\hat{B}C$ criterion. Moreover, the 1-median subgraph may contain circles (as the one discussed in fig. 4). Since the G_{Host} subgraph may include nodes that lie far from the current $Host$, the algorithm in the MP case exhibits extra agility to reach faster the final service location. However, in the evaluation of the practical cDSMA implementation it has been shown that the spatially bounded G_{Host} subgraph of the SP case only marginally increases the hopcount values compared to the ones of MP.

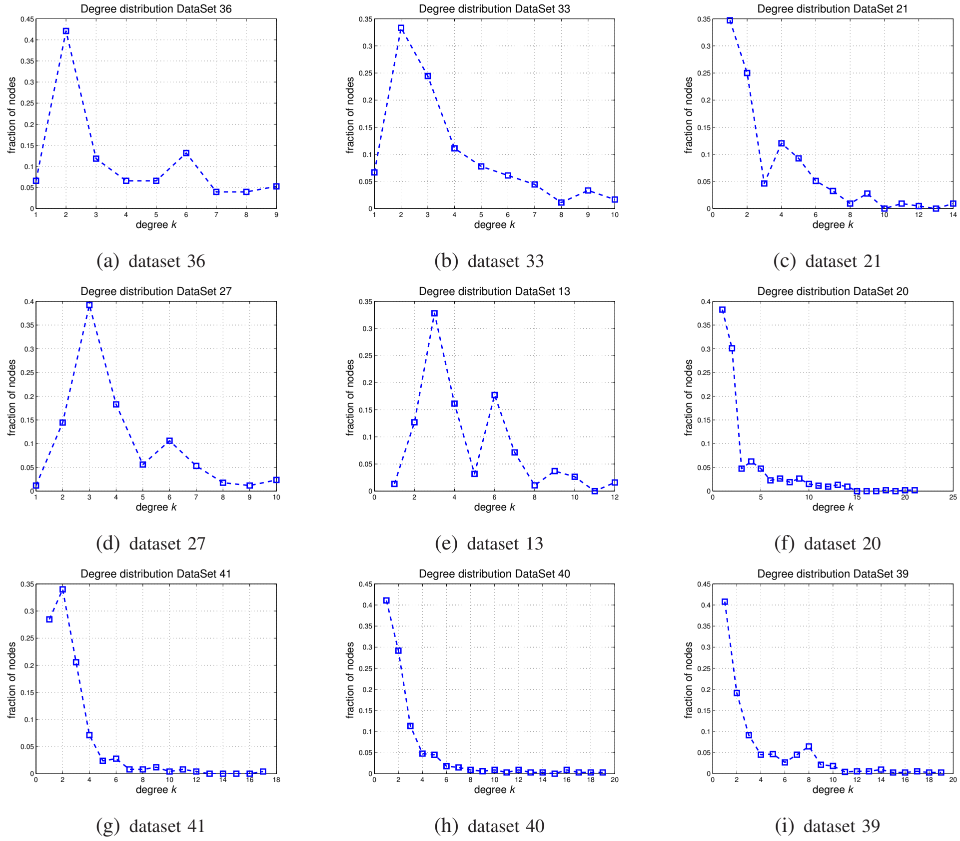
Next, we show how the topological information collected through these dedicated messages *suffices* for carrying out the demand mapping task, even when more than one shortest-path is utilized by the employed routing protocol.

3.3 Global demand mapping on the 1-median subgraph.

After the derivation of the 1-median subgraph, the current service host needs to further process the $\alpha|V|$ measurement-reporting messages that correspond to the selected subgraph nodes. The way this will be done depends as well on the deployed routing protocol.

Under multipath routing each network node u bears sets of weight factors $\{wf_{uj}(u;t)\}$ expressing what portion of the traffic destined for node t is routed over the outgoing link $\{u, j\}$, where j is a neighbor node. Clearly, under the SP routing strategy the weight factor sets $\{wf_{uj}(u;t)\}$ are singleton, their single element being unity.

The mapping task under MP routing is more complicated than SP routing. Each node u now sends *one measurement-reporting message per shortest path* used towards the current service host node (hereafter, superscript k enumerates the different msg_x^k messages that the host receives from x). Besides the measured traffic load values, $wC\hat{B}C(u;t)$, and the nodes lying on the path, the message logs the

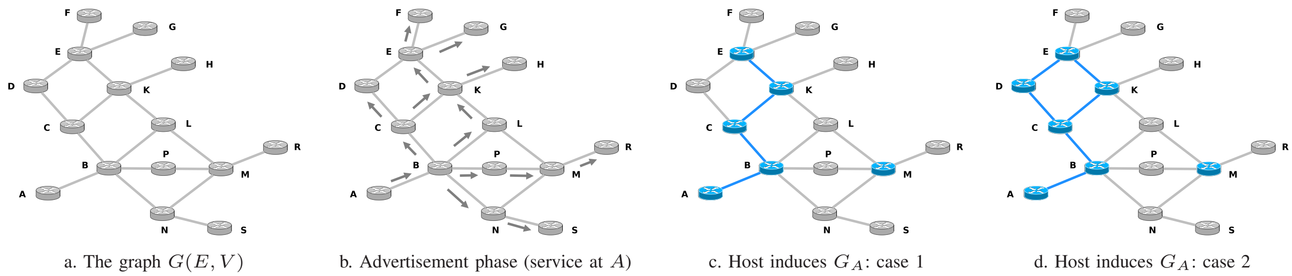


(a) dataset 36 (b) dataset 33 (c) dataset 21

(d) dataset 27 (e) dataset 13 (f) dataset 20

(g) dataset 41 (h) dataset 40 (i) dataset 39

Fig. 3. Degree distributions of an indicative subset of the ISP topology snapshots.



a. The graph $G(E, V)$ b. Advertisement phase (service at A) c. Host induces G_A: case 1 d. Host induces G_A: case 2

Fig. 4. Example of cDSMA protocol implementation under uniform randomized MP routing. The G_A is non-connected for case 1 when each node exhibits a unity of demand except for E,L and M that have $w(E) = 4$, $w(L) = 0$ and $w(M) = 5$, respectively. The case 2 under which the G_A contains a cycle, differentiates from case 1 only in that $w(D) = 2$.

corresponding weight factor $wf_{u,j}(u;t)$ of the traffic routed to t with node j as next hop as well as the respective weight factors of the nodes lying on the path to t. As earlier explained the 1-median subgraph may now deviate from the connected tree topology; it is hereafter treated as a hierarchy H_{Host}, whereby each node is annotated with its depth d with respect to $Host$, with $d(Host) = 0$. When processing a node z in a given msg_x^k, all nodes logged deeper(outer) in the message are called ancestors (descendants) of z in msg_x^k.

The Algorithm that carries out the demand mapping is called DeMaMP and presented in 1. Like its counterpart for single-path routing, DeMaMP sequentially parses only the measurement-reporting messages of the 1-median subgraph nodes (*selected* nodes) in decreasing length order of their msg_x^k part, the one logging the path nodes. It initializes the nodes' w_{eff} variables to the measured traffic values and then seeks to subtract all traffic demand contributions that have already been credited to nodes further up the hierarchy.

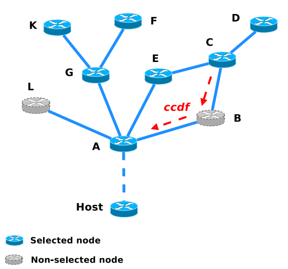
However, when compared to DeMaSP operation, there are two main discrepancies resulting from the different structure of the 1-median subgraph. Firstly, the set of logged nodes within a msg_x^k includes both nodes that have been selected in the 1-median subgraph and nodes that have not been (*e.g.*, in fig. 5-left, msg_D^2 contains the non-selected node B). While treating a node z within a msg_x^k, the algorithm distinguishes whether it is selected or not. To correctly discount (*e.g.*, from node A in msg_D^2) the

Algorithm 1 Message header parsing and demand mapping under MP (DeMaMP)

input: *set of selected nodes in* G_{Host},
  $\{msg_u\}\ \forall u \in G_{Host}$
output: *vector* $w_{eff}(u)\ \forall u \in G_{Host}$

Initialization
for all $x \in G_{Host}$ **do** $w_{eff}(x) = w\hat{CBC}(x)$
vector $B \leftarrow$ *sort all* msg_x *in decreasing order of* $|msg_x|$

for $i = 1$ **up to** $Len(B)$ **do**
  parse $B(i) = msg_x$
  int $cddf = 0$;
  for $m = 1$ **up to** $|msg_x(m))|$ **do**
    if $m == 1$
      $last\_sel = msg_x(1)$
      $cddf = rf_{msg_x(1), msg_x(2)}(msg_x(1); Host)$
    else
    if $m \leq |msg_x(m))| - 1$
      $l = msg_x(m),\ n = msg_x(m+1)$
      if $l \in G_{Host}$
        if $path[last\_sel \rightarrow l]\ != marked$
          $w_{eff}(l) = w_{eff}(l) - w\hat{CBC}(last\_sel) * cddf$
          mark $path[last\_sel \rightarrow l]$ *as read*
          $cddf = rf_{l,n}(l; Host)$
        end if
        $last\_sel = l$
      else
        $cddf = cddf * rf_{l,n}(l; Host)$
      endif
    else// *mapping on host*
      $h = msg_x(m)$
      if $path[last\_sel \rightarrow h]\ != marked$
        $w_{eff}(h) = w_{trans}(h; Host) + w(h) - w\hat{CBC}(last\_sel) * cddf$
        mark $path[last\_sel \rightarrow h]$ *as read*
      endif
    endif
  end for
end for



Header of

msg^1_D:	D	C	E	A	..Host
msg^2_D:	D	C	B	A	..Host
msg^1_K:		K	G	A	..Host
msg^1_F:		F	G	A	..Host
msg^1_C:		C	E	A	..Host
msg^2_C:		C	B	A	..Host
msg^1_E:			E	A	..Host
msg^1_G:			G	A	..Host
msg^1_A:				A	..Host

Fig. 5. **Right:** Demand mapping under MP: $w_{eff}(A) = w_{map}(A) + w(A)$. The message header parsing process must reveal to the $Host$ that the $w_{mapp}(A)$ equals the cumulative demand of the non-selected L and B nodes that communicate with the $Host$ through A. That is $w_{eff}(A) = w\hat{CBC}(A) - (w\hat{CBC}(G) + w\hat{CBC}(E) + cddf)$. **Left:** Message headers received by the $Host$ and sorted in decreasing length for parsing.

traffic demand that has already been credited to another selected node, the one first encountered while flowing towards the current service host (*i.e.*, node C), DeMaMP uses the *cumulative demand discount factor* (cddf) variable. The latter stores anytime, for a given parsed message, the amount of traffic of the earlier selected node that flows through the path encoded in the message and, thus, has to be discounted by the first next selected node (*i.e.*, node A); clearly, cddf equals the corresponding weight factor when the earlier selected node is z's immediate ancestor.

Secondly, if z is a selected node in msg^k_x (*e.g.*, node C in message msg^1_D), it no longer holds that $msg^k_z \subset msg^k_u$. Therefore, we cannot discard whole messages upon reading their first entry and it does not suffice to check whether node z has been processed; we rather need to know whether the full path from the last selected node encountered over the msg^k_x, to z *i.e.*, $path[last\_sel \rightarrow z]$ has been earlier taken into account. Similar care is needed for the $w_e ff(Host)$ computation which is -contrary to DeMaSP- carried out by directly parsing[1] the *last* entry of each msg^k_x.

3.4 1-median solution within the G_{Host} subgraph

This step is carried out in the same way regardless the employed routing protocol.

4 Study of Overload Phenomena

In this Section we aim to investigate whether the cDSMA operation may lead to overload phenomena in the presence of multiple service instances across the network. Overload phenomena include a) placing a significant number of services over a small number of highly central nodes; and/or b) routing increased aggregate demand traffic towards the services' final locations through one or few more nodes. In what follows, we generate realistic scenarios for the demand distribution of multiple available services and experimentally show that cDSMA does not give rise to any overload phenomena of either type.

4.1 Spatial concentration of service instances

cDSMA will accumulate a number of services to the same final location in the extreme theoretical scenario that the demand distribution over the network users is identical for each single service made available in the network. On the contrary, more realistic service demand scenarios cater for differentiation of its values across services and network users.

To emulate such realistic demand distributions for every available service, we use the following model. We recognize the different popularity among the services and retain the assumption of Zipf access patterns, namely the cumulative demand value is Zipf distributed across the different service instances. We then need to determine a) the number N_{Rq} of requester nodes for each service; b) which are those N_{Rq} requesters out of the total N nodes; and finally c) what part of the total demand generated for each service is assigned to every corresponding requester.

1. Alternatively, we can compute the $w_e ff(Host)$ value by subtracting the demand that we assign over the $G_{Host} \setminus Host$ subgraph from the total network demand. The later quantity though may not be available in case we seek to employ exclusively local information. Such an option is considered when we study the performance of the cDSMA practical implementations (Section 8).

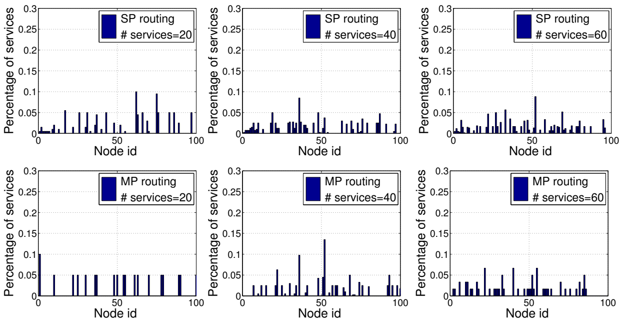


Fig. 6. Percentage of services placed by cDSMA per node for Dataset 35 under SP and MP routing.

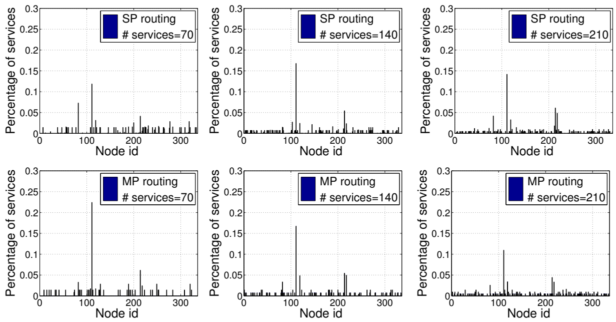


Fig. 7. Percentage of services placed by cDSMA per node for Dataset 40 under SP and MP routing.

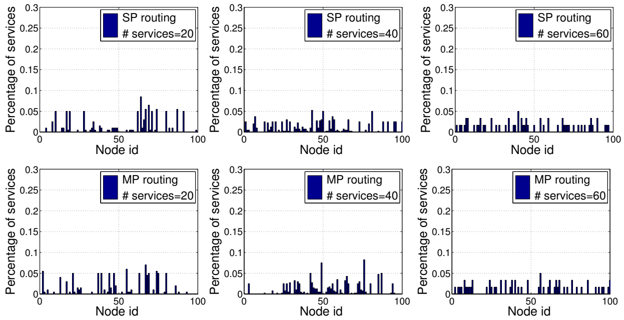


Fig. 8. Percentage of services placed by cDSMA per node for a 10x10 grid network under SP and MP routing.

To determine N_{Rq} we employ a heuristic way; it aims to reflect that the higher the rank k of a service, the more the nodes that are expected to request it. As such, when there are NoS available services in a network of N nodes, then the requesters for the service of rank k are given by:

$$N_{Rq} = N(1 - \frac{\delta \cdot k}{NoS}) \tag{2}$$

where δ is a constant that can fine-tune the percentage of N_{Rq} over N, set for our experiments to 0.9. We randomly choose the set of N_{Rq} nodes out of the total network nodes and assign to each of them an amount of service demand following a power-law distribution. The latter choice models the differentiation of the demand for a certain service among its users.

In figures 6, 7 and 8 we plot the percentage of services that cDSMA places over each node of two real-world (*i.e.*, datasets 35, 40) and one synthetic topology (*i.e.*, 10x10 grid), respectively (these plots are chosen as representative ones for discussion, we got similar results for all our topologies). We experiment with both the single- (SP) and multi-path (MP) routing strategies keeping the G_{Host} subgraph size equal to 7 for the dataset 35 and the grid, and 9 for the dataset 40. Table 4 in the main paper suggest that these values yield a normalized cost of no more than 1.022 for both routing options in real world topologies and a slightly higher one for the grid topology, as already expected from our proof-of-concept study in Section 6 of the main paper. Finally, we scale the number of services from approximately 0.2 up to 0.6 of the total number of nodes.

The results in figures 6, 7 and 8 show that cDSMA correctly identifies the demand gradient for each single service and finally manages to distribute the instances across the network nodes, avoiding to overload the central ones. With no significant differences between the two routing options as well as the scaling of the services' number, the nodes in dataset 35 and the grid appear to be almost uniformly sharing the burden of service hosting; even the most central ones in topological terms *i.e.*, the nodes 52, 48 and 36 for the former and 45, 46, 55 and 56 for the latter, host no more than 10% of the available services.

In the dataset 40 (Fig. 7), nodes 111 and 214 seem to consistently attract an increased number of services compared to the rest of the network nodes; yet, only in one case does their number correspond to as much as the 20% of the total number of services. Both these nodes, *i.e.*, top and third top in terms of betweenness centrality (BC), attain extremely large centrality values compared to the rest and express a topological bias that benefits concentration phenomena despite the service demand differentiation across services and users. In that sense, the variance of the original BC metric or even the ratio between its maximum and mean value across the network can roughly reflect the potential for service concentration. We measure the BC variance equal to 3.3×10^4 for the grid topology, while the topologies D35 and D40 yield about 3 and 180 times as much, respectively. This finding is well reflected when comparing the above figures; the depicted percentages clearly suggest that even under unfavorable network topologies that avail a few highly central nodes, cDSMA keeps low levels of service concentration for both routing strategies.

4.2 Traffic routing overhead

In this section we turn our attention to the impact of the final services' locations on the amount of demand traffic each node needs to route. Intuitively, when services tend to spatially concentrate, the nearby nodes would need to bear heavy routing duties as the traffic somehow converges to the service locations. The results of the previous section show that cDSMA achieves, under realistic demand dynamics,

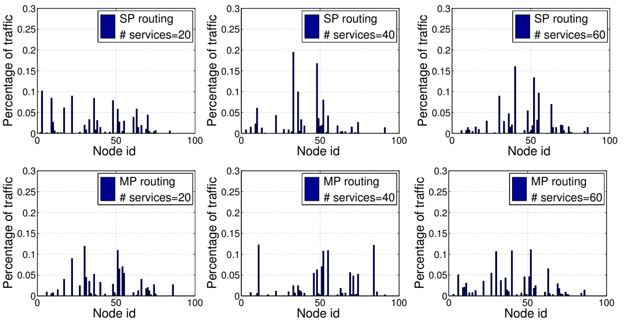


Fig. 9. Percentage of demand-traffic each node routes after cDSMA completion for Dataset 35 under SP and MP routing.

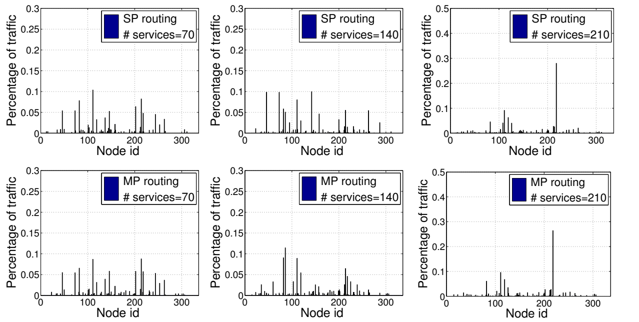


Fig. 10. Percentage of demand-traffic each node routes after cDSMA completion for Dataset 40 under SP and MP routing.

the spatial distribution of services among multiple hosts and therefore avoids any concentration phenomena. Here, we aim to assess whether the traffic load each node bears is also favorably distributed among the nodes. To this end, we retain our previous choices as to how the demand for each available service emerges across the network users and experimentally measure the demand traffic that each node routes after the cDSMA has completed the placement of all available services.

Figures 9, 10 and 11 present what percentage of the total demand traffic served by the network, each node routes towards the locations of the corresponding services under SP and MP routing. Clearly, each plot corresponds to the service placement configuration depicted earlier, in figures 6, 7 and 8, respectively. In figure 9 where the topological bias is not intense (see the previous subsection), it seems that the routing affects the traffic load more significantly than it did earlier with the service concentration; inline with intuition, under the single-path (SP) routing choice some nodes, in most cases the central ones, end up routing more traffic than others, even if there is no node having received many services (*e.g.*, the case with 40 services). Yet, this traffic never exceeds the 20% percentage. MP, on the other hand, spreads the traffic over many paths and elevates many different nodes up to lower percentages. Finally, our results confirm that the service concentration at some node increases the routing load on the nearby nodes. In the MP case with 40 services, there is a slight concentration over nodes 52 and 36, whereas in the corresponding traffic measurement we find that the 4 nodes that route about 10% of the traffic each *i.e.*, 11, 52, 55 and 84, do lie from 1 to 3 hops away the concentration locations.

Regarding the dataset 40 (Fig. 9), the network topology avails some highly central locations (*i.e.*, nodes 214, 111) that end up being traversed by increased amounts of demand traffic under both routing strategies. Still, the corresponding percentages are in all cases kept below 30%. In the same spirit, the grid topology (Fig. 11) is not characterized by such a sharp topological contrast and accordingly helps the cDSMA operation to spread the total demand traffic amount across multiple nodes. As a final comment,

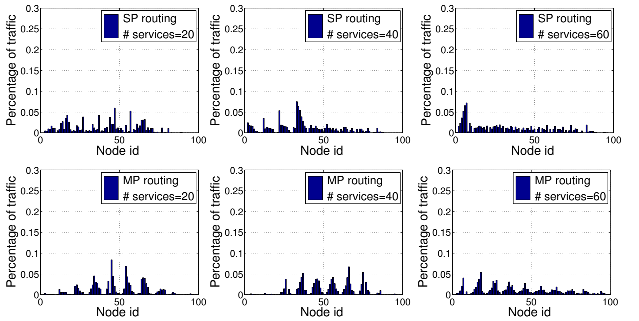


Fig. 11. Percentage of demand-traffic each node routes after cDSMA completion for a 10x10 grid topology under SP and MP routing.

note that the mean normalized cost that cDSMA yields over the real-world network topologies remains below than 1.02 (for the selected G_{Host} sizes). This means that even in the ideal scenario of having an *optimal distributed* algorithm in place, the resulted service placement configuration would be roughly the same with the one derived by cDSMA; as such, the ISP network topology would also need to tolerate similar values of demand traffic overhead.

5 DATA REPLICATION AND PLACEMENT RELATED LITERATURE

Data replication [2] refers to the storage of files or, more generally, information objects, in specific points in a network, so that they can be retrieved by requesting nodes at smaller access costs. Earlier research in this area has mostly considered centralized implementations [3], [4] of file placements.

More recently, the networking community has turned its attention to replica (*i.e.*, server or object) placement schemes over the Internet, devised to facilitate the efficient content distribution. The placement of *service* facilities where we focus on, is usually studied along the same thread although sometimes may exhibit different characteristics such as the absence of replication (which renders it more difficult to tackle). With that in mind, we consider works that seek to improve the Content Distribution Networks (CDNs) performance through heuristic solutions for server placement quite close to ours. In [5] the authors adopt a k-median formulation of the problem and propose several heuristic solutions. The Greedy algorithm sequentially places one replica at a time; the current one is placed at the lowest-cost location exhaustively determined under the assumptions that a) the so far placed replicas remain fixed b) a node's requests are directed to the closest replica. It has been shown to achieve placements within a factor of 1.1-1.5 of the optimal for synthetic and real-world network topologies under demand patterns extracted from server load traces. Less effective (*i.e.*, ratio is between 1.6-2) is the Hot Spot heuristic that places the replicas at the top k nodes that along with their vicinity generate the greatest load.

HotZone, a latency-based variant of the Hot Spot heuristic has been proposed in [6]. The authors employ a system that models the Internet as a M-dimensional space and estimates the latency between two nodes as the distance between their corresponding coordinates. First, they identify k groups of nodes whose latency is relatively low and rank them according to the demand load they generate. Then the one with the minimum average distance in each group is chosen as the replica-holding node. The approach is shown to produce comparable results to the above heuristics while maintaining lower complexity. Finally, in [7] the authors investigate the replica placement problem from a QoS standpoint. They seek to minimize the storage and update cost of all candidate servers with a replication strategy that satisfies the QoS requirements in terms of object retrieval cost. The optimal QoS-aware placement problem is shown to be NP-complete when the nodes are aware of the adopted replication strategy and access their closest replica. Greedy heuristics are then introduced while there exist polynomial optimal solutions, otherwise.

Adopting a *centralized* approach renders the above heuristics irrelevant to the networking environment we consider and consequently excludes them from serving the purposes of a fair and meaningful comparison with cDSMA. A decentralized solution yet not within the typical facility location framework amounts to viewing the problem as graph coloring [8]. Each node is characterized by a color representing a content class and updates the information about its nearby nodes' colors via a modified Bellman-Ford algorithm. Asynchronously, a node seeks to change its color such as to maximize the distance to a fellow node with the same color. The algorithm converges to colorings where the distance from a node to an arbitrarily chosen color is bounded by a factor of three compared to the optimal colorings.

In view of the emerging autonomic environments where individual nodes may act selfishly, a recent Internet content distribution thread involves placement approaches that employ game-theoretic arguments. The distributed selfish replication game is introduced and studied in [9], where the authors propose an algorithm for its solution and analyze its main properties. In [9] the assumption is that all nodes within a group can communicate and cooperate with each other. More recently, Pacifici and Dan in [10] relax this assumption and consider replication games over arbitrary social graphs, which introduce constraints on the possible interaction patterns between the players. They derive sufficient conditions for letting the players reach an equilibrium of the game and propose a distributed algorithm in this respect. On the other hand, Borst *et al.* in [11] assume altruistic players making placements that maximize the aggregate benefit over the whole network rather than theirs. The performance of their greedy algorithm is within a constant factor of two from the globally optimal performance under arbitrary demands and, even closer under identical content preferences and uniform cache capacities.

Finally, data replication has also been studied in the context of mobile social networks, with social characteristics being embedded into data replication algorithms. In [12], the authors construct a dynamic learning algorithm where nodes from various social communities opt for a utility-maximizing content placement strategy based on their encounters with other nodes. The content utility is related to the availability of content in different communities, as well as the ties a user has with each community. In [13] the authors study how content is distributed in an opportunistic network considering both technical constraints (*e.g.*, battery/processing power and wireless bandwidth) and user preferences. In [14] the authors propose an approach that can enhance content dissemination by associating both interest- and locality-based dynamics of social groups.

References

[1] Hong Shen, "Fast path-disjoint routing in transputer networks", *Microprocessing and Microprogramming*, vol.33, no.1, pp 21-31, 1991.

[2] T. Loukopoulos, I. Ahmad and D. Papadias, "An Overview of Data Replication on the Internet" , *Proc. Int'l Symposium on Parallel Architectures, Algorithms and Networks*, Washington, USA, 2002.

[3] A Leff, J.L. Wolff and P.S. Yu, "Replication algorithms in a remote caching architecture", *IEEE Trans. Par. and Distr. Systems*, vol.4, no.11, pp 1185-1204, 1993.

[4] W.W. Chu, "Optimal File Allocation in a Multiple Computer System", *IEEE Transactions on Computers*, vol. 18, no. 10, pp 885-889, 1969

[5] L. Qiu, V. Padmanabhan, and G. Voelker, "On the Placement of Web Server Replicas", *Proc. IEEE INFOCOM*, 2001, Anchorage, Alaska.

[6] G. Pierre, M. Szymaniak and M. van Steen, "Latency-Driven Replica Placement (extended version)", *IPSJ Journal*, Vol. 47, No. 8, 2006.

[7] X. Tang and J. Xu, "QoS-Aware Replica Placement for Content Distribution", *IEEE Trans. Par. and Distr. Systems*, vol.16, no.10, pp 921-932, 2005.

[8] B.J. Ko and D. Rubenstein, "Distributed self-stabilizing placement of replicated resources in emerging networks", *IEEE/ACM Trans. on Netw.*, vol.13, no.3, pp. 476- 487, 2005.

[9] N. Laoutaris, O. Telelis, V. Zissimopoulos and I. Stavrakakis, "Distributed selfish replication", *IEEE Trans. Par. Distr. Systems*, vol.17, no.12, pp 1401-1413, 2006.

[10] V. Pacifici and G. Dán, "Selfish content replication on graphs", *Proc. 23rd Int'l Teletraffic Congress*, 2011, San Francisco, CA, USA.

[11] S. Borst, V. Gupta and A. Walid, "Distributed caching algorithms for content distribution networks", *Proc. 29th IEEE Infocom*, 2010, San Diego, California, USA.

[12] C. Boldrini, M. Conti and A. Passarella, "Design and performance evaluation of ContentPlace, a social-aware data dissemination system for opportunistic networks", *Computer Networks*, vol. 54, no. 4, pp 589-604, 2010.

[13] I. Carreras, D. Tacconi and A. Bassoli, "Social Opportunistic Computing: Design for Autonomic User-Centric Systems", *Autonomic Communication*, Springer, 2010.

[14] E. Jaho and I. Stavrakakis, "Joint interest- and locality-aware content dissemination in social networks", *Proc. 6th Int'l IEEE/IFIP WONS*, 2009, Snowbird, Utah, USA.