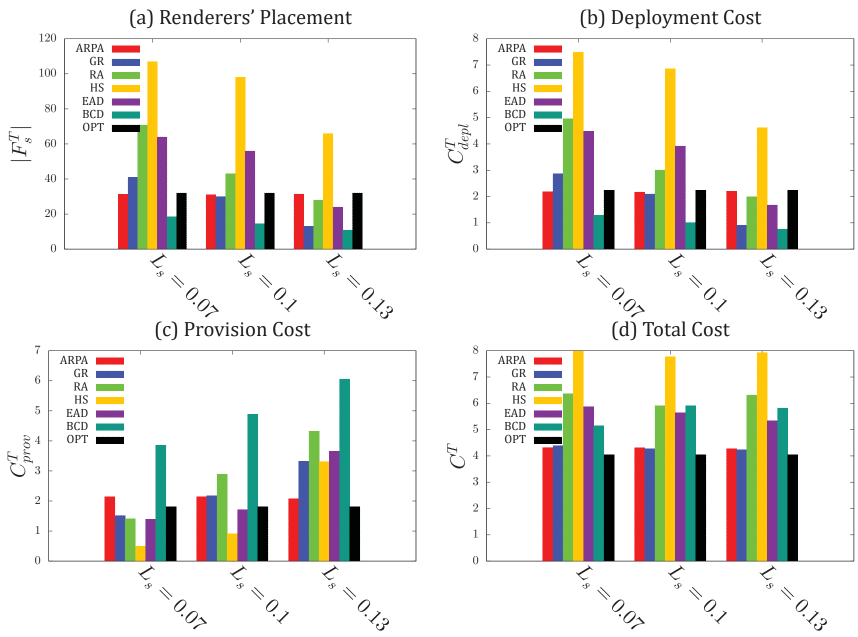

Fig. 9 – Comparisons with fixed $C_s^{host} = 0.07$ and $\xi = 0.2$, for three values of the upper latency bound L_s.

for $L_s = \{0.1, 0.13\}$, resulting in a margin of $\simeq 2\%$ and 37.52% over ARPA, respectively. HS presets the highest factor of increment which is coupled with the greatest reduction rate in the number of renderers, as stated previously. RA and EAD also yield gradually higher C_{prov}^T with the former having a substantially larger percentage of growth. For the latter, it is noteworthy that for $L_s = 0.1$ it manages to approximate OPT better than any alternative, which however is in accordance with the second largest placement as observed previously. Yet, L_s again has a negative impact on EAD for higher values because, as it expands, EAD's properties allow hub renderers (i.e., those that are closest to the center of the denser network sites) to progressively satisfy more distant fog nodes, which however are served now with an analogous increment in their render cost. BCD, as expected, generates the highest provision cost in all cases which additionally worsens as multimedia latency tolerance becomes greater, a fact that is perfectly coupled with its ability to produce steadily smaller $|F_s^T|$. This indicates its suitability for minimizing the deployment cost, which however is not weighed against the loss in service provisioning.

Summarizing both deployment and provision costs, in Fig. 9(d) we plot the total cost acquired by each algorithm. Clearly, GR and ARPA outperform the alternatives by a great magnitude (with a solution at most 9% away from OPT), with the former yielding slightly better results (a performance gap of less than 2% on average from ARPA). Still, different than GR which is a purely centralized algorithm that employs global knowledge to approximate OPT, ARPA is a completely decentralized solution. Therefore, ARPA's high approximation ability can be exceedingly leveraged in dynamic, mobile, or large-scale multimedia systems where changes in network conditions frequently occur, and so continuous exhaustive searches, such as the ones executed by GR, are not viable in the long run in terms of computational complexity and cost. EAD, following an overall affinity to reduce the C^T as L_s grows, seems to be the next best option when considering that HS repeatedly fails to minimize the total cost, while BCD presents the opposite trend with a tendency to increase C^T (note, here, that lower L_s values favor BCD). Finally, RA remains impartial offering poor performance under any L_s. The aforementioned highlight the superiority of ARPA in addressing the objective function of LB-UFRL, i.e., the trade-off between both the gain and the loss of the overall rendering service deployment and provision costs, and in the meanwhile provide strict multimedia delay guarantees that are obeyed under all circumstances to offer maximum QoS.

In regards to Set #3 and to increase fairness, we only provide insights regarding the behavior of ARPA compared to OPT and omit all other algorithms since they do not factor in during their execution the value of C_s^{host}. Fig. 10 encases the obtained results under fixed $\xi = 0.2$ and $L_s = 0.08$. The particular values are intentionally set to low in order to observe how ARPA reacts and adjusts the placement after the delay has been compensated for weakly-connected networks. Apparently, from Fig. 10(a) and 10(b) we can remark that ARPA achieves a good approximation of OPT in terms of C_{depl}^T, managing a deviation of 11.17% from OPT when $C_s^{host} = 0.08$, while for lesser values, i.e., when $C_s^{host} = 0.06$ and $C_s^{host} = 0.04$, its performance gap is reduced to 6.34% and 3.12% respectively, since more replications take place and service consolidation can transpire more easily. The opposite trends are revealed by Fig. 10(c) where the margin for C_{prov}^T between the two algorithms tends to increase, which is actually to be expected when in combination with the previous

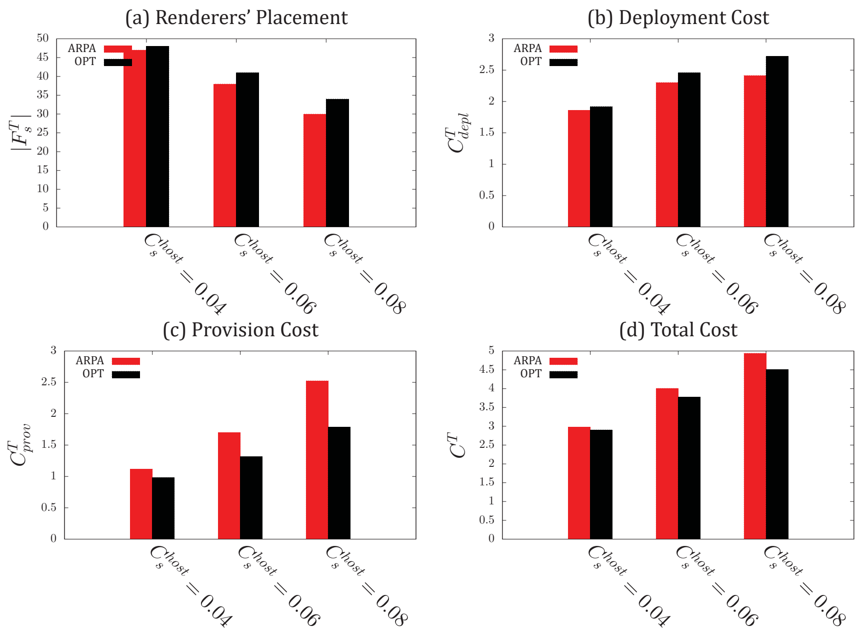

Fig. 10 – Comparisons against the optimal solution with fixed $L_s = 0.08$ and $\xi = 0.2$, for three values of the host cost C_s^{host}.

findings and the accelerated reduction rate in the renderers (i.e., as host cost grows) in which case the provision gain from deploying more rendering services becomes cost-unworthy for the provider. Despite the negative impact on service provisioning, the total cost posed by ARPA remains relatively close to OPT, as reported by Fig. 10(d). In fact, even for the worst case (i.e, for $C_s^{host} = 0.08$) ARPA still manages to sufficiently converge towards the optimal solution with a divergence of 9.64% from OPT, while for a lesser host cost, i.e., $C_s^{host} = 0.06$ and $C_s^{host} = 0.04$, the performance gap reduces to 6.13% and 2.55% accordingly, indicating that the prevention rate of ARPA's relocation properties in respect to replication and consolidation decreases significantly, allowing for an augmentation in the final placement.

The last claim, along with all previous observations made, are validated further by Table 2, which provides the average number of occurrences for each relocation rule (i.e., migration, replication, and consolidation) for all conducted experimental sets and scenarios. For Set #1 it is obvious that higher values of ξ favor fewer relocations since the rise in connectivity radius allows for faster convergence. Noteworthy is the fact that for $\xi = 0.15$, the migration frequency more than triples that compared to $\xi = 0.35$ since, as already mentioned, the number of links is significantly smaller which in turn necessitates more service movements to lead the renderers to suitable locations within the FAN. For the same reason, consolidation is commended for lower values of ξ, because longer distances render inefficient the operation of extra renderers (i.e., Rule 4). For Set #2 we can remark that as the multimedia delay tolerance increases, fewer replications are required to compensate for L_s (i.e., Rule 2). As a result of the lesser number of replicas present, the rate of consolidation also drops, hence, migration takes over to optimize the placement. Finally, regarding Set #3 we observe that a higher host cost indeed halts the replication process significantly (i.e., Rule 3). As such, consolidation and migration also experience a decay (since fewer replica candidates satisfy ARPA's relocation Rules 1 and 4), a fact that explains the growth in the overall service provision cost.

Table 2 – ARPA's average number of rendering service relocations per experimental set and scenario.

Set	Scenario	Migrations	Replications	Consolidations
#1	$\xi = 0.15$	20.6	43.6	3.8
	$\xi = 0.25$	8.8	42.4	2.9
	$\xi = 0.35$	6.1	41.8	1.9
#2	$L_s = 0.07$	9.6	32.5	3.1
	$L_s = 0.10$	9.7	31.7	2.5
	$L_s = 0.13$	10	31.5	2.3
#3	$C_s^{host} = 0.04$	10.8	49.5	4.9
	$C_s^{host} = 0.06$	9.3	38.3	1.8
	$C_s^{host} = 0.08$	8.2	29.8	1.5

10. CONCLUSION

In this paper, the problem of efficient placement of renderers within fog-assisted cloud multimedia computing systems was addressed and studied. Based on the cloud/fog architecture, the goal was to locate rendering services at key locations within the fog layer, in order to achieve timely service provision with simultaneous deployment cost minimization, meeting stringent delay guarantees regarding the expected 3D rendering time.

To address the particular problem, a constrained optimization approach was followed to mathematically formulate the problem within a finite time horizon. Given its high complexity, a thorough theoretical analysis was provided that designates the minimum acceptable conditions for relocating the services towards optimal locations, based solely on localized network information, that iteratively lead to total cost reduction during each discrete time slot until a solution is found. Subsequently, a distributed algorithm, namely the ARPA, capitalizing on the analysis was proposed and discussed. The properties of ARPA allow the renderers to autonomously migrate, replicate, or complementarily merge, until an optimal placement has been reached.

Comprehensive simulations under various deployment scenarios, in accordance with the analytical findings, confirmed the algorithm's expected behavior and proved its efficiency for fast convergence and delay guarantees, even when fluctuations appear regarding the network conditions during runtime. As such, ARPA was shown to be resilient to changes and capable of scaling up and down to meet the current demands, producing placement solutions that closely approximate the optimal one. Through additional trace-driven evaluations, we also showcased ARPA's robustness and applicability on realistic topologies. The results clearly exhibited its efficacy in outperforming past approaches and in generating near-optimal placements that rival even the solutions of the purely centralized greedy approach.

Future research will involve ARPA's real-world deployment using appropriate test beds. Besides, no relocation cost was assumed here during the placement, and no capacities were reported in the model. Thus, in future work, the impact of such dependencies will also be investigated.

A. APPENDIX

Here you may find the proofs of the various lemmas and theorems in their order of appearance within the paper.

A.1 Proof of Lemma 1

When migration takes place at a given time slot, the renderers' placement changes for the next time slot. Therefore, there exists a cost difference between the two consecutive time slots attributed to the new $\mathcal{SP}$s that are formed. However, to calculate this difference would require global topology knowledge. Thus, assuming that ${}^{t}\overrightarrow{x_s : y}$ occurs during the time slot t and the service s has moved *instantaneously* to the new location within $\delta(x_s) \cap S^t_{x_s}$, then there exists a new *M-hypothetical total cost*, denoted as $\overrightarrow{C^t}$, incurred by the new *M-hypothetical placement*, denoted as $\overrightarrow{F^t_s}$, after migration. Then, the render cost $\forall u \in S^t_{x_s}$ is also affected when taking into account the fact that their $r^t(u)$ are accumulated along the established $\mathcal{SP}$ and up to the previous x_s. On the contrary, the remainder fog nodes of the FAN (i.e., $F^t_s \backslash S^t_{x_s}$) remain practically unaffected by this change, since they are served by different renderers that do not contribute additional cost differences to the $\overrightarrow{C^t}$ when their $\mathcal{SP}$s remain unaltered during the instantaneous migration. Let $y \in \delta(x_s) \cap S^t_{x_s}$ be the neighbor node where $d(x_s, y) > d(x_s, z), \forall z \in \delta(x_s) \cap S^t_{x_s}$ (i.e., the worst-case scenario). Consequently, the minimum post-migration cost difference can now be expressed as $\Delta C({}^{t}\overrightarrow{x_s : y})_{min} = C^t - C^{t+1} = C^t - \overrightarrow{C^t} = C^t_{prov} + C^t_{depl} - \overrightarrow{C^t_{prov}} - \overrightarrow{C^t_{depl}}$. However, since $|F^t_s| = |\overrightarrow{F^t_s}|$ then $C^t_{depl} = \overrightarrow{C^t_{depl}}$ and the previous statement can now be written as $\Delta C({}^{t}\overrightarrow{x_s : y})_{min} = \sum_{\forall u \in S^t_{x_s}} C^t_{ren}(x_{s\to u}) - \sum_{\forall u \in S^t_{y_s}} \overrightarrow{C^t_{ren}}(y_{s\to u})$. However, from Eq. *(3)* we have $\Delta C({}^{t}\overrightarrow{x_s : y})_{min} = \sum_{\forall u \in S^t_{x_s}} r^t(u) D(u, x_s) - \sum_{\forall u \in S^t_{y_s}} r^t(u) D(u, y_s)$, or equivalently $\Delta C({}^{t}\overrightarrow{x_s : y})_{min} = C^t_{acc}(x_s) - \overrightarrow{C^t_{acc}}(y_s)$, and eventually $\Delta C({}^{t}\overrightarrow{x_s : y})_{min} = C^t_{acc}(x_s) - C^t_{acc}(y)$.

A.2 Proof of Theorem 1

Based on Lemma 1 we have $\Delta C({}^{t}\overrightarrow{x_s : y})_{min} = C^t_{acc}(x_s) - C^t_{acc}(y)$. To determine if $\Delta C({}^{t}\overrightarrow{x_s : y}) > 0, \forall y \in \delta(x_s) \cap S^t_{x_s}$, and in turn if $C^t > C^{t+1}$, it is sufficient enough to show when $\Delta C({}^{t}\overrightarrow{x_s : y})_{min} > 0 \Leftrightarrow C^t_{acc}(x_s) - C^t_{acc}(y) > 0 \Leftrightarrow C^t_{acc}(x_s) > C^t_{acc}(y) \Leftrightarrow C^t_{acc}(x_s) > \overrightarrow{C^t_{acc}}(y_s)$. Note that $C^t_{acc}(x_s) = \sum_{\forall u \in S^t_{x_s}} r^t(u) D(u, x_s) = \sum_{\forall u \in S^t_{x_s} \backslash \check{S}^t_{x_s:y}} r^t(u) D(u, x_s) + \sum_{\forall u \in \check{S}^t_{x_s:y}} r^t(u) D(u, y) + R^t(y) d(y, x_s)$. Likewise, $\overrightarrow{C^t_{acc}}(y_s) = \sum_{\forall u \in S^t_{y_s}} r^t(u) D(u, y_s) = \sum_{\forall u \in S^t_{y_s} \backslash \check{S}^t_{y_s:x}} r^t(u) D(u, y_s) + \sum_{\forall u \in \check{S}^t_{y_s:x}} r^t(u) D(u, x) + \overrightarrow{R^t}(x) d(x, y_s)$, where $\overrightarrow{R^t}(x)$ are the *M-hypothetical aggregate rendering demands* of fog node x after the migration. By subtracting $\overrightarrow{C^t_{acc}}(y_s)$ from $C^t_{acc}(x_s)$ we get $C^t_{acc}(x_s) - \overrightarrow{C^t_{acc}}(y_s) = \sum_{\forall u \in S^t_{x_s} \backslash \check{S}^t_{x_s:y}} r^t(u) D(u, x_s) + \sum_{\forall u \in \check{S}^t_{x_s:y}} r^t(u) D(u, y) + R^t(y) d(y, x_s) - \sum_{\forall u \in S^t_{y_s} \backslash \check{S}^t_{y_s:x}} r^t(u) D(u, y_s) - \sum_{\forall u \in \check{S}^t_{y_s:x}} r^t(u) D(u, x) - \overrightarrow{R^t}(x) d(x, y_s)$. Because $d(y, x_s) = d(x, y_s) = d(x, y)$, and since $S^t_{x_s} \backslash \check{S}^t_{x_s:y} = \check{S}^t_{y_s:x}$ and $\check{S}^t_{x_s:y} = S^t_{y_s} \backslash \check{S}^t_{y_s:x}$ (otherwise the nodes will have preferred a closer renderer with less delay which definitely leads to total cost decay), then $C^t_{acc}(x_s) - \overrightarrow{C^t_{acc}}(y_s) = R^t(y) d(x, y) - \overrightarrow{R^t}(x) d(x, y) = (R^t(y) - \overrightarrow{R^t}(x)) d(x, y)$. Therefore, we require $\Delta C({}^{t}\overrightarrow{x_s : y})_{min} = (R^t(y) - \overrightarrow{R^t}(x)) d(x, y) > 0$. Since by default $d(x, y) > 0$ then for $\Delta C({}^{t}\overrightarrow{x_s : y})_{min} > 0$ it is enough for $R^t(y) - \overrightarrow{R^t}(x) > 0$. However, $\overrightarrow{R^t}(x) = R^t(x_s) - R^t(y)$, and so $R^t(y) - \overrightarrow{R^t}(x) = R^t(y) - (R^t(x_s) - R^t(y)) = R^t(y) - R^t(x_s) + R^t(y) > 0 \Leftrightarrow \Delta C({}^{t}\overrightarrow{x_s : y})_{min} = 2R^t(y) - R^t(x_s) > 0$. This is a lower bound condition for the minimum post-migration cost difference in order to reach total cost reduction, and so considering the general case, it can be inferred that for any $y \in \delta(x_s) \cap S^t_{x_s}$ if $\Delta C({}^{t}\overrightarrow{x_s : y}) \geq 2R^t(y) - R^t(x_s) > 0$ then $C^t - C^{t+1} > 0$ is also satisfied.

A.3 Proof of Lemma 2

Since service replication ${}^{t}\overleftrightarrow{x_s : y'}$ will create a new renderer at the fog node $y' \in V$, then it is ensured that $F^{t+1}_s = F^t_s \cup \{y'_s\}$ and so $C^t_{depl} < C^{t+1}_{depl}$. Obviously, fog nodes where $D(u, y'_s) < D(u, x_s), \forall u \in V, \forall x_s \in F_s^{t+1} \backslash y'_s$ will prefer to be served by the former renderer, i.e., y'_s. Thus, it is trivial to conclude that in this case $C_{ren}^t(x_{s \to u}) > C_{ren}^{t+1}(y'_{s \to u})$, or $C_{acc}^t(x_s) > C_{acc}^{t+1}(y'_s)$, and in turn $C_{prov}^t > C_{prov}^{t+1}$.

A.4 Proof of Theorem 2

First, when ${}^t\overleftrightarrow{x_s : y'}$ then it is ensured that $F_s^{t+1} = F_s^t \cup \{y'_s\}$, and thus $C^t = C^{t+1} + \Delta C({}^t\overleftrightarrow{x_s : y'})$. Second, in order for $C^t > C^{t+1}$ we require that $\Delta C({}^t\overleftrightarrow{x_s : y'}) > 0$. Still, in view of Lemma 2 it is impossible to know beforehand how C^{t+1} will be affected by the $\mathcal{SP}$ alterations caused by the replication, since this would require global topology knowledge. Instead, assuming that ${}^t\overleftrightarrow{x_s : y'}$ happens *instantaneously* during t and no $\mathcal{SP}$ recreations occur that will surely increase the $\Delta C({}^t\overleftrightarrow{x_s : y'})$, then there is an *R-hypothetical total cost*, denoted as $\overleftrightarrow{C^t}$, incurred by the new *R-hypothetical placement*, denoted as $\overleftrightarrow{F_s^t}$, after replication. Then, the post-replication cost difference becomes $\Delta C({}^t\overleftrightarrow{x_s : y'}) = C^t - \overleftrightarrow{C^t}$. Note that via Eq. *(8)* the $C^t = C_{prov}^t + C_{depl}^t = \sum_{\forall v_s \in F_s^t} \sum_{\forall u \in S_{v_s}^t} C_{ren}^t(v_{s \to u}) + \sum_{\forall v_s \in F_s^t} C_s^{host} = \sum_{\forall v_s \in F_s^t} \sum_{\forall u \in S_{v_s}^t} r^t(u) D(u, v_s) + |F_s^t| C_s^{host}$. Because our primary interest is on the particular renderer that participates in the replication process, then this relation is equivalent to writing $C^t = \sum_{\forall v_s \in F_s^t \backslash \{x_s\}} \sum_{\forall u \in S_{v_s}^t} r^t(u) D(u, v_s) + |F_s^t \backslash \{x_s\}| C_s^{host} + \sum_{\forall u \in S_{x_s}^t} r^t(u) D(u, x_s) + C_s^{host}$. Likewise, $\overleftrightarrow{C^t} = \overleftrightarrow{C_{prov}^t} + \overleftrightarrow{C_{depl}^t} = \sum_{\forall v_s \in \overleftrightarrow{F_s^t}} \sum_{\forall u \in S_{v_s}^t} \overleftrightarrow{C_{ren}^t}(v_{s \to u}) + \sum_{\forall v_s \in \overleftrightarrow{F_s^t}} C_s^{host} = \sum_{\forall v_s \in \overleftrightarrow{F_s^t}} \sum_{\forall u \in S_{v_s}^t} r^t(u) D(u, v_s) + |\overleftrightarrow{F_s^t}| C_s^{host}$. However, considering that $\overleftrightarrow{F_s^t} = F_s^t \cup \{y'\}$, then $\overleftrightarrow{C^t} = \sum_{\forall v_s \in F_s^t \backslash \{x_s\}} \sum_{\forall u \in S_{v_s}^t} r^t(u) D(u, v_s) + |F_s^t \backslash \{x_s\}| C_s^{host} + \sum_{\forall u \in \overleftrightarrow{S_{x_s}^t}} r^t(u) D(u, x_s) + C_s^{host} + \sum_{\forall u \in \overleftrightarrow{S_{y'_s}^t}} r^t(u) D(u, y'_s) + C_s^{host}$. Thus, by subtracting the two costs we quickly arrive to the result $\Delta C({}^t\overleftrightarrow{x_s : y'}) = \sum_{\forall u \in S_{x_s}^t} r^t(u) D(u, x_s) - \sum_{\forall u \in \overleftrightarrow{S_{x_s}^t}} r^t(u) D(u, x_s) - \sum_{\forall u \in \overleftrightarrow{S_{y'_s}^t}} r^t(u) D(u, y'_s) - C_s^{host} \Leftrightarrow \Delta C({}^t\overleftrightarrow{x_s : y'}) = \sum_{\forall u \in S_{x_s}^t \backslash \check{S}_{x_s:y'}^t} r^t(u) D(u, x_s) + \sum_{\forall u \in \check{S}_{x_s:y'}^t} r^t(u) D(u, y') + R^t(y') d(y', x_s) - \sum_{\forall u \in \overleftrightarrow{S_{x_s}^t}} r^t(u) D(u, x_s) - \sum_{\forall u \in \overleftrightarrow{S_{y'_s}^t}} r^t(u) D(u, y'_s) - C_s^{host}$. Since $S_{x_s}^t \backslash \check{S}_{x_s:y'}^t = \overleftrightarrow{S_{x_s}^t}$ and $\check{S}_{x_s:y'}^t = \overleftrightarrow{S_{y'_s}^t}$, i.e., they are identical, then ultimately $\Delta C({}^t\overleftrightarrow{x_s : y'}) = R^t(y') d(y', x_s) - C_s^{host}$. Subsequently, for $C^t > \overleftrightarrow{C^t}$ it is enough $\Delta C({}^t\overleftrightarrow{x_s : y'})_{min} = \{\Delta C({}^t\overleftrightarrow{x_s : y'}) : R^t(y') d(y', x_s) \leq R^t(z) d(z, x_s), \forall z \in \delta(x_s) \cap S_{x_s}^t\} > 0$ (i.e., the worst-case scenario). The last is a lower bound condition for the minimum post-replication cost difference in order to achieve total cost reduction, and so considering the general case, for any $y' \in \delta(x_s) \cap S_{x_s}^t$, if $\Delta C({}^t\overleftrightarrow{x_s : y'}) \geq R^t(y') d(y', x_s) - C_s^{host} > 0$ then $C^t - C^{t+1} > 0$ is also satisfied.

A.5 Proof of Lemma 3

Since service consolidation ${}^t\overleftarrow{x_s : y''_s}$ will merge two renderers, then it is ensured that $F_s^{t+1} = F_s^t \backslash \{y''_s\}$ and so $C_{depl}^t > C_{depl}^{t+1}$. Then, fog nodes where $D(u, y''_s) < D(u, x_s), \forall u \in V, \forall x_s \in F_s^t$ will for $t = t + 1$ be served by a new renderer, i.e., some $x_s \in F_s^{t+1}$. Thus, it is trivial to conclude that in this case $C_{ren}^{t+1}(x_{s \to u}) > C_{ren}^t(y''_{s \to u})$, or $C_{acc}^t(x_s) > C_{acc}^{t+1}(y'_s)$, and in turn $C_{prov}^t < C_{prov}^{t+1}$.

A.6 Proof of Theorem 3

The proof is similar to that of Theorem 2. When ${}^t\overleftarrow{x_s : y''_s}$ then it is ensured that $F_s^{t+1} = F_s^t \backslash \{y''_s\}$, and thus $C^t = C^{t+1} + \Delta C({}^t\overleftarrow{x_s : y''_s})$. Second, in order for $C^t > C^{t+1}$ we necessitate that $\Delta C({}^t\overleftarrow{x_s : y''_s}) > 0$. Yet, in view of Lemma 3 it is impossible to predict how C^{t+1} will be affected since this would demand global knowledge. Instead, assuming that ${}^t\overleftarrow{x_s : y''_s}$ happens *instantaneously* during t and no $\mathcal{SP}$ reformations happen, then there is a *C-hypothetical total cost*, denoted as $\overleftarrow{C^t}$, due to the new *C-hypothetical placement*, denoted as $\overleftarrow{F_s^t}$, after consolidation takes place. Like migration, we also demand here that $d(y''_s, x_s) > d(y''_s, v_s), \forall v_s \in F_s^t \cap \delta(y''_s)$ (i.e., the worst-case scenario), otherwise the rendering demands of all $u \in S_{y''_s}^t$ will for $t = t + 1$ be forwarded to other renderers with less rendering costs, which in turn will further increase $\Delta C({}^t\overleftarrow{x_s : y''_s})$. Thus, to obtain the lower bound we must enforce that x_s is the worst candidate for the consolidation. Then, the minimum post-consolidation cost difference becomes $\Delta C({}^t\overleftarrow{x_s : y''_s})_{min} = C^t - \overleftarrow{C^t}$. Through Eq. *(8)* the $C^t = C_{prov}^t + C_{depl}^t = \sum_{\forall v_s \in F_s^t} \sum_{\forall u \in S_{v_s}^t} C_{ren}^t(v_{s \to u}) + \sum_{\forall v_s \in F_s^t} C_s^{host} = \sum_{\forall v_s \in F_s^t} \sum_{\forall u \in S_{v_s}^t} r^t(u) D(u, v_s) + |F_s^t| C_s^{host} = \sum_{\forall v_s \in F_s^t \backslash \{x_s, y''_s\}} \sum_{\forall u \in S_{v_s}^t} r^t(u) D(u, v_s) + |F_s^t \backslash \{x_s, y''_s\}| C_s^{host} + \sum_{\forall u \in S_{x_s}^t} r^t(u) D(u, x_s) + \sum_{\forall u \in S_{y''_s}^t} r^t(u) D(u, y''_s) + 2C_s^{host}$. Likewise, $\overleftarrow{C^t} = \overleftarrow{C_{prov}^t} + \overleftarrow{C_{depl}^t} = \sum_{\forall v_s \in \overleftarrow{F_s^t}} \sum_{\forall u \in S_{v_s}^t} \overleftarrow{C_{ren}^t}(v_{s \to u}) + \sum_{\forall v_s \in \overleftarrow{F_s^t}} C_s^{host} = \sum_{\forall v_s \in \overleftarrow{F_s^t}} \sum_{\forall u \in S_{v_s}^t} r^t(u) D(u, v_s) + |\overleftarrow{F_s^t}| C_s^{host}$. However, since $\overleftarrow{F_s^t} = F_s^t \backslash \{y''_s\}$, then $\overleftarrow{C^t} = \sum_{\forall v_s \in F_s^t \backslash \{x_s, y''_s\}} \sum_{\forall u \in S_{v_s}^t} r^t(u) D(u, v_s) + |F_s^t \backslash \{x_s, y''_s\}| C_s^{host} + \sum_{\forall u \in \overleftarrow{S_{x_s}^t}} r^t(u) D(u, x_s) + C_s^{host} = \sum_{\forall v_s \in F_s^t \backslash \{x_s, y''_s\}} \sum_{\forall u \in S_{v_s}^t} r^t(u) D(u, v_s) + |F_s^t \backslash \{x_s, y''_s\}| C_s^{host} + \sum_{\forall u \in S_{x_s}^t} r^t(u) D(u, x_s) + \sum_{\forall u \in S_{y''_s}^t} r^t(u) D(u, y''_s) + R^t(y''_s) d(y''_s, x_s) + C_s^{host}$. Thus, by subtracting the two costs, $\Delta C({}^t\overleftarrow{x_s : y''_s})_{min} = \sum_{\forall u \in S_{x_s}^t} r^t(u) D(u, x_s) + \sum_{\forall u \in S_{y''_s}^t} r^t(u) D(u, y''_s) + 2C_s^{host} - \sum_{\forall u \in S_{x_s}^t} r^t(u) D(u, x_s) - \sum_{\forall u \in S_{y''_s}^t} r^t(u) D(u, y''_s) - R^t(y''_s) d(y''_s, x_s) - C_s^{host} \Leftrightarrow$

$\Delta C(\overleftrightarrow{{}^t x_s : y''_s})_{min} = C_s^{host} - R^t(y''_s)d(y''_s, x_s) > 0$. This is a lower bound condition for the minimum post-consolidation cost difference to achieve total cost reduction, and so regarding the general case $\forall y''_s \in \delta(x_s)$ it is proved that if $\Delta C(\overleftrightarrow{{}^t x_s : y''_s}) \geq C_s^{host} - R^t(y''_s)d(y''_s, x_s) > 0$ then $C^t - C^{t+1} > 0$ is also satisfied.

A.7 Proof of Lemma 4

Our focus is on the worst-case scenario where the underlying $\mathcal{SP}$s are not affected by the replications. Let $u \in S_{v_s}^t$ represent a fog node where $C_{ren}^t(v_{s \to u}) > L_s$ and let $\hat{C}_{ren}^t(v_s) = \max\{\hat{C}_{ren}^t(\nu)_{\forall \nu \in S_{v_s}^t \cap \delta(v_s)}\} = C_{ren}^t(v_{s \to u})$. Obviously, when $\overleftrightarrow{{}^t v_s : y'}$ occurs, then for $t = t+1$ the $C_{acc}^t(x_s) > C_{acc}^{t+1}(\nu_s) \Leftrightarrow C_{ren}^t(v_{s \to u}) > C_{ren}^t(\nu_{s \to u})$. By iteratively replicating services down the $\mathcal{SP}$ connecting the initial v_s to u the render cost of u will continue to drop. This process will repeat until the render cost becomes less than L_s, or in the extreme occasion when u itself becomes a renderer in which case $C_{ren}^{t+k_{u,v_s}}(u_{s \to u}) = r^t(u)D(u, u_s) = r^t(u)d(u, u)$. Since $d(u, u) = 0$, then $C_{ren}^{t+k_{u,v_s}}(u_{s \to u}) = 0$. Thus, it will require at the most k_{u,v_s} replications for $C_{ren}^t(v_{s \to u}) < L_s$ to be satisfied.

A.8 Proof of Theorem 4

This is easily proved by contradiction. Assume that for some arbitrary t, the placement process has been terminated and some node z still exists in the FAN where $C_{ren}^t(\omega_{s \to z}) > L_s$ ($\omega_s \in F_s^t$ and $k_{z,\omega_s} \geq 1$). Then, $\hat{C}_{ren}^t(\omega_s) > L_s$ is also satisfied. Thus, renderer ω_s will immediately initiate a replication process to compensate for L_s. This leads to a contradiction since it is implied that the placement process has not yet ended. Actually, according to Lemma 4 and given that (i) each fog node is served by exactly one renderer (i.e., the one with the minimum render cost), and (ii) no other alterations will appear in the underlying $\mathcal{SP}$s, then it will additionally require at most $t = t + k_{z,\omega_s}$ time slots for the relocation process to terminate. Thus, at the end of the rendering service relocation, no fog node exists that violates L_s.

A.9 Proof of Lemma 5

ARPA's execution is iterative; that is, each renderer per t will attempt to relocate its service according to one (and exactly one) of the rules provided, by sequentially validating their conditions. In the worst case this means traversing though all neighbor nodes of a fog renderer with maximum neighborhood size. Thus, the time complexity is bounded, i.e., $\mathcal{O}(|F_s^t|N)$, where $N \triangleq \max_{\forall u \in V} |\delta(u)|$.

A.10 Proof of Theorem 5

It is suffice to show that there cannot exist loops, i.e., repeated visits to the same configuration of renderers within the FAN. We distinguish four cases depending on the L_s and C_s^{host} that capture all the different relocation alternatives.

α. The initial placement of renderers, i.e., the F_s^0 abides by the L_s constraint. Then, the LB-UFRL effectively reduces to a pure UFL problem where the services relocate in a loop-free manner amongst the fog nodes of G. This means that the placement, following a monotonically decreasing total cost path based on Rules 1, 3 and 4, will visit each combination of potential renderers only once. Since the solution space is finite, the ARPA will terminate after a finite number of time slots, i.e., when the $C^t > C^{t+1}$ criterion ceases to apply or when the ARPA reaches the optimal solution of the LB-UFRL.

β. The initial placement of renderers does not abide by the L_s constraint. Then after the compensation of the L_s is completed via Rule 2, the ARPA will adopt the same behavior as described previously and ultimately converge to a solution where $C^t \leq C^{t+1}$ for all other relocation rules, in which case it will also terminate.

γ. The C_s^{host} is set to a very high value prohibiting the opening of new replicas, i.e., the execution of Rule 3. Then we can further observe two cases. First, following the L_s compensation we will have k renderers placed in the FAN. Thus, the LB-UFRL effectively reduces to a pure k-median problem and the ARPA will reach in a finite number of time slots through Rules 1 and 4 to a solution with up to k renderers present in the FAN. Second, if the initial placement consists of only one renderer, i.e., $|F_s^0| = 1$ and the L_s bound is not infringed, i.e., no other replications or consolidations can occur, the LB-UFRL is transformed to a 1-median problem. Even in this case, the migration criterion in Rule 1 will lead the ARPA to a solution, given the total cost being monotonically decreasing between successive time slots. Thus, the ARPA will again terminate.

δ. The C_s^{host} is set to a very low value which prohibits the merging of services. Once more, after replicas are created (i.e., due to Rules 2 or 3), then Rule 1, because it is executed first, will always migrate the services towards optimized locations that reduce the total cost. Hence, ARPA will converge to a solution and terminate in finite time slots.

ACKNOWLEDGEMENT

This work was supported in part by project "Corfu Virtual Exhibition Site for Tourism-Culture-Environment" (MIS 5031252), which is partially funded by European and National Greek Funds (ESPA) under the Regional Operational Programme "Ionian Islands 2014-2020".

REFERENCES

[1] Subashini Subashini and Veeraruna Kavitha. "A survey on security issues in service delivery models of cloud computing". In: *Journal of network and computer applications* 34.1 (2011), pp. 1–11.

[2] Wenwu Zhu, Chong Luo, Jianfeng Wang, and Shipeng Li. “Multimedia cloud computing”. In: *IEEE Signal Processing Magazine* 28.3 (2011), pp. 59–69.

[3] Wei Cai, Ryan Shea, Chun-Ying Huang, Kuan-Ta Chen, Jiangchuan Liu, Victor CM Leung, and Cheng-Hsin Hsu. “A survey on cloud gaming: Future of computer games”. In: *IEEE Access* 4 (2016), pp. 7605–7620.

[4] Tim Verbelen, Pieter Simoens, Filip De Turck, and Bart Dhoedt. “Leveraging cloudlets for immersive collaborative applications”. In: *IEEE Pervasive Computing* 12.4 (2013), pp. 30–38.

[5] Ryan Shea, Di Fu, and Jiangchuan Liu. “Cloud gaming: Understanding the support from advanced virtualization and hardware”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 25.12 (2015), pp. 2026–2037.

[6] Zheng Xue, Di Wu, Jian He, Xiaojun Hei, and Yong Liu. “Playing high-end video games in the cloud: A measurement study”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 25.12 (2014), pp. 2013–2025.

[7] Yichao Jin, Yonggang Wen, and Kyle Guan. “Toward cost-efficient content placement in media cloud: Modeling and analysis”. In: *IEEE Transactions on Multimedia* 18.5 (2016), pp. 807–819.

[8] Qiumin Lu, Jianguo Yao, Zhengwei Qi, Bingsheng He, et al. “Fairness-efficiency allocation of cpu-gpu heterogeneous resources”. In: *IEEE Transactions on Services Computing* (2016).

[9] Ivan Slivar, Mirko Suznjevic, and Lea Skorin-Kapov. “Game Categorization for Deriving QoE-Driven Video Encoding Configuration Strategies for Cloud Gaming”. In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14.3s (2018), pp. 1–24.

[10] Hao Chen, Xu Zhang, Yiling Xu, Ju Ren, Jingtao Fan, Zhan Ma, and Wenjun Zhang. “T-Gaming: A Cost-Efficient Cloud Gaming System at Scale”. In: *IEEE Transactions on Parallel and Distributed Systems* 30.12 (2019), pp. 2849–2865.

[11] Mohaddeseh Basiri and Abbas Rasoolzadegan. “Delay-aware resource provisioning for cost-efficient cloud gaming”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 28.4 (2016), pp. 972–983.

[12] Yongqiang Gao, Lin Wang, and Jiantao Zhou. “Cost-Efficient and Quality of Experience-Aware Provisioning of Virtual Machines for Multiplayer Cloud Gaming in Geographically Distributed Data Centers”. In: *IEEE Access* 7 (2019), pp. 142574–142585.

[13] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B Letaief. “A survey on mobile edge computing: The communication perspective”. In: *IEEE Communications Surveys & Tutorials* 19.4 (2017), pp. 2322–2358.

[14] Yunhua Deng, Yusen Li, Ronald Seet, Xueyan Tang, and Wentong Cai. “The server allocation problem for session-based multiplayer cloud gaming”. In: *IEEE Transactions on Multimedia* 20.5 (2017), pp. 1233–1245.

[15] Yuhua Lin and Haiying Shen. “CloudFog: leveraging fog to extend cloud gaming for thin-client MMOG with high quality of service”. In: *IEEE Transactions on Parallel and Distributed Systems* 28.2 (2017), pp. 431–445.

[16] Sharon Choy, Bernard Wong, Gwendal Simon, and Catherine Rosenberg. “A hybrid edge-cloud architecture for reducing on-demand gaming latency”. In: *Multimedia systems* 20.5 (2014), pp. 503–519.

[17] Ping Lu, Quanying Sun, Kaiyue Wu, and Zuqing Zhu. “Distributed online hybrid cloud management for profit-driven multimedia cloud computing”. In: *IEEE Transactions on Multimedia* 17.8 (2015), pp. 1297–1308.

[18] Athanasios Tsipis, Konstantinos Oikonomou, Vasileios Komianos, and Ioannis Stavrakakis. “Performance evaluation in cloud-edge hybrid gaming systems”. In: *Third International Balkan Conference on Communications and Networking 2019 (BalkanCom’19)*. Skopje, North Macedonia. 2019.

[19] Nimrod Megiddo and Kenneth J Supowit. “On the complexity of some common geometric location problems”. In: *SIAM journal on computing* 13.1 (1984), pp. 182–196.

[20] Pitu B Mirchandani and Richard L Francis. *Discrete location theory*. 1990.

[21] Atakan Aral and Tolga Ovatman. “A decentralized replica placement algorithm for edge computing”. In: *IEEE transactions on network and service management* 15.2 (2018), pp. 516–529.

[22] Mohammad A Salahuddin, Jagruti Sahoo, Roch Glitho, Halima Elbiaze, and Wessam Ajib. “A survey on content placement algorithms for cloud-based content delivery networks”. In: *IEEE Access* 6 (2017), pp. 91–114.

[23] Giuseppe Portaluri, Davide Adami, Andrea Gabbrielli, Stefano Giordano, and Michele Pagano. “Power consumption-aware virtual machine placement in cloud data center”. In: *IEEE Transactions on Green Communications and Networking* 1.4 (2017), pp. 541–550.

[24] Meng Wang, Xiaoqiao Meng, and Li Zhang. "Consolidating virtual machines with dynamic bandwidth demand in data centers". In: *2011 Proceedings IEEE INFOCOM*. IEEE. 2011, pp. 71–75.

[25] Shvan Omer, Sadoon Azizi, Mohammad Shojafar, and Rahim Tafazolli. "A priority, power and traffic-aware virtual machine placement of IoT applications in cloud data centers". In: *Journal of Systems Architecture* 115 (2021), p. 101996.

[26] Ao Zhou, Shangguang Wang, Bo Cheng, Zibin Zheng, Fangchun Yang, Rong N Chang, Michael R Lyu, and Rajkumar Buyya. "Cloud service reliability enhancement via virtual machine placement optimization". In: *IEEE Transactions on Services Computing* 10.6 (2016), pp. 902–913.

[27] Abdulaziz Alashaikh, Eisa Alanazi, and Ala Al-Fuqaha. "A Survey on the Use of Preferences for Virtual Machine Placement in Cloud Data Centers". In: *ACM Computing Surveys (CSUR)* 54.5 (2021), pp. 1–39.

[28] Mohammad Goudarzi, Huaming Wu, Marimuthu S Palaniswami, and Rajkumar Buyya. "An Application Placement Technique for Concurrent IoT Applications in Edge and Fog Computing Environments". In: *IEEE Transactions on Mobile Computing* (2020).

[29] Lili Qiu, Venkata N Padmanabhan, and Geoffrey M Voelker. "On the placement of web server replicas". In: *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*. Vol. 3. IEEE. 2001, pp. 1587–1596.

[30] Sin-Shuen Cheung and David P Williamson. "Greedy algorithms for the single-demand facility location problem". In: *Operations Research Letters* 45.5 (2017), pp. 452–455.

[31] Kun Cao, Liying Li, Yangguang Cui, Tongquan Wei, and Shiyan Hu. "Exploring placement of heterogeneous edge servers for response time minimization in mobile edge-cloud computing". In: *IEEE Transactions on Industrial Informatics* 17.1 (2020), pp. 494–503.

[32] Thomas Moscibroda and Rogert Wattenhofer. "Facility location: distributed approximation". In: *Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*. 2005, pp. 108–117.

[33] Jagruti Sahoo, Mohammad A Salahuddin, Roch Glitho, Halima Elbiaze, and Wessam Ajib. "A survey on replica server placement algorithms for content delivery networks". In: *IEEE Communications Surveys & Tutorials* 19.2 (2016), pp. 1002–1026.

[34] Shangguang Wang, Jinliang Xu, Ning Zhang, and Yujiong Liu. "A survey on service migration in mobile edge computing". In: *IEEE Access* 6 (2018), pp. 23511–23528.

[35] Mirsaeid Hosseini Shirvani, Amir Masoud Rahmani, and Amir Sahafi. "A survey study on virtual machine migration and server consolidation techniques in DVFS-enabled cloud datacenter: taxonomy and challenges". In: *Journal of King Saud University-Computer and Information Sciences* 32.3 (2020), pp. 267–286.

[36] Changchun Long, Yang Cao, Tao Jiang, and Qian Zhang. "Edge computing framework for cooperative video processing in multimedia IoT systems". In: *IEEE Transactions on Multimedia* 20.5 (2017), pp. 1126–1139.

[37] Hua-Jun Hong, De-Yu Chen, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. "Placing virtual machines to optimize cloud gaming experience". In: *IEEE Transactions on Cloud Computing* 3.1 (2014), pp. 42–53.

[38] Alexandros Doumanoglou, David Griffin, Javier Serrano, Nikolaos Zioulis, Truong Khoa Phan, David Jiménez, Dimitrios Zarpalas, Federico Alvarez, Miguel Rio, and Petros Daras. "Quality of expe- rience for 3-D immersive media streaming". In: *IEEE Transactions on Broadcasting* 64.2 (2018), pp. 379–391.

[39] Tuo Cao, Zhuzhong Qian, Kun Wu, Mingxian Zhou, and Yibo Jin. "Service Placement and Bandwidth Allocation for MEC-enabled Mobile Cloud Gaming". In: *2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE. 2021, pp. 179–188.

[40] Adyson M Maia, Yacine Ghamri-Doudane, Dario Vieira, and Miguel F de Castro. "Optimized place- ment of scalable iot services in edge computing". In: *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE. 2019, pp. 189–197.

[41] Jonghwa Choi and Sanghyun Ahn. "Optimal Service Provisioning for the Scalable Fog/Edge Computing Environment". In: *Sensors* 21.4 (2021), p. 1506.

[42] Yusen Li, Yunhua Deng, Xueyan Tang, Wentong Cai, Xiaoguang Liu, and Gang Wang. "Cost-efficient server provisioning for cloud gaming". In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14.3s (2018), pp. 1–22.

[43] Amira Rayane Benamer, Nejib Ben Hadj-Alouane, and Khaled Boussetta. "Online Games Servers Placement in Fog Computing: an Hybrid Bio-inspired Approach". In: *2020 IEEE 45th LCN Symposium on Emerging Topics in Networking (LCN Symposium)*. IEEE. 2020, pp. 141–149.

[44] Yuan Zhang, Lei Jiao, Jinyao Yan, and Xiaojun Lin. "Dynamic Service Placement for Virtual Reality Group Gaming on Mobile Edge Cloudlets". In: *IEEE Journal on Selected Areas in Communications* 37.8 (2019), pp. 1881–1897.

[45] Derian Alencar, Cristiano Both, Rodolfo Antunes, Helder Oliveira, Eduardo Cerqueira, and Denis Rosário. "Dynamic Microservice Allocation for Virtual Reality Distribution with QoE support". In: *IEEE Transactions on Network and Service Management* (2021).

[46] Redowan Mahmud, Satish Narayana Srirama, Kotagiri Ramamohanarao, and Rajkumar Buyya. "Quality of Experience (QoE)-aware placement of applications in Fog computing environments". In: *Journal of Parallel and Distributed Computing* 132 (2019), pp. 190–203.

[47] Mohammad Goudarzi, Marimuthu Palaniswami, and Rajkumar Buyya. "A distributed application placement and migration management techniques for edge and fog computing environments". In: *2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE. 2021, pp. 37–56.

[48] Paridhika Kayal and Jörg Liebeherr. "Autonomic service placement in fog computing". In: *2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE. 2019, pp. 1–9.

[49] Yuanzhe Li, Ao Zhou, Xiao Ma, and Shangguang Wang. "Profit-aware Edge Server Placement". In: *IEEE Internet of Things Journal* (2021).

[50] Ashkan Yousefpour, Ashish Patil, Genya Ishigaki, Inwoong Kim, Xi Wang, Hakki C Cankaya, Qiong Zhang, Weisheng Xie, and Jason P Jue. "FOGPLAN: A lightweight QoS-aware dynamic fog service provisioning framework". In: *IEEE Internet of Things Journal* 6.3 (2019), pp. 5080–5096.

[51] Tero Lähderanta, Teemu Leppänen, Leena Ruha, Lauri Lovén, Erkki Harjula, Mika Ylianttila, Jukka Riekki, and Mikko J Sillanpää. "Edge computing server placement with capacitated location allocation". In: *Journal of Parallel and Distributed Computing* 153 (2021), pp. 130–149.

[52] Lu Zhao, Wenan Tan, Bo Li, Qiang He, Li Huang, Yong Sun, Lida Xu, and Yun Yang. "Joint Shareability and Interference for Multiple Edge Application Deployment in Mobile Edge Computing Environment". In: *IEEE Internet of Things Journal* (2021).

[53] Palash Roy, Sujan Sarker, Md Abdur Razzaque, Mohammad Mehedi Hassan, Salman A AlQahtani, Gianluca Aloi, and Giancarlo Fortino. "AI-enabled mobile multimedia service instance placement scheme in mobile edge computing". In: *Computer Networks* 182 (2020), p. 107573.

[54] Feifei Chen, Jingwen Zhou, Xiaoyu Xia, Hai Jin, and Qiang He. "Optimal application deployment in mobile edge computing environment". In: *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*. IEEE. 2020, pp. 184–192.

[55] Lorenzo Corneo, Nitinder Mohan, Aleksandr Zavodovski, Walter Wong, Christian Rohner, Per Gunningberg, and Jussi Kangasharju. "(How Much) Can Edge Computing Change Network Latency?" In: *2021 IFIP Networking Conference (IFIP Networking)*. IEEE. 2021, pp. 1–9.

[56] Feng Zeng, Yongzheng Ren, Xiaoheng Deng, and Wenjia Li. "Cost-effective edge server placement in wireless metropolitan area networks". In: *Sensors* 19.1 (2019), p. 32.

[57] C-A La, Pietro Michiardi, Claudio Casetti, C-F Chiasserini, and Marco Fiore. "A lightweight distributed solution to content replication in mobile networks". In: *2010 IEEE Wireless Communication and Networking Conference*. IEEE. 2010, pp. 1–6.

[58] Sharrukh Zaman and Daniel Grosu. "A distributed algorithm for the replica placement problem". In: *IEEE Transactions on Parallel and Distributed Systems* 22.9 (2011), pp. 1455–1468.

[59] Panagiotis Pantazopoulos, Merkouris Karaliopoulos, and Ioannis Stavrakakis. "Distributed placement of autonomic internet services". In: *IEEE transactions on parallel and distributed systems* 25.7 (2013), pp. 1702–1712.

[60] Eleni Kavvadia, Spyros Sagiadinos, Konstantinos Oikonomou, Giorgos Tsioutsiouliklis, and Sonia Aïssa. "Elastic virtual machine placement in cloud computing network environments". In: *Computer Networks* 93 (2015), pp. 435–447.

[61] Ryan Shea, Jiangchuan Liu, Edith C-H Ngai, and Yong Cui. "Cloud gaming: architecture and performance". In: *IEEE network* 27.4 (2013), pp. 16–21.

[62] Athanasios Tsipis, Konstantinos Oikonomou, Vasileios Komianos, and Ioannis Stavrakakis. "QoE-Aware Rendering Service Allocation in Fog-Assisted Cloud Gaming Environments". In: *2020 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*. IEEE. 2020, pp. 1–8.

[63] Guy Almes, Sunil Kalidindi, Matthew Zekauskas, and Al Morton. "A one-way delay metric for IP performance metrics (IPPM)". In: *IETF, January* 10 (2016).

[64] Klervie Toczé, Johan Lindqvist, and Simin Nadjm-Tehrani. "Performance study of mixed reality for edge computing". In: *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*. 2019, pp. 285–294.

[65] András Varga and Rudolf Hornig. “An overview of the OMNeT++ simulation environment”. In: *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*. 2008, pp. 1–10.

[66] Phu Lai, Qiang He, Mohamed Abdelrazek, Feifei Chen, John Hosking, John Grundy, and Yun Yang. “Optimal edge user allocation in edge computing with variable sized vector bin packing”. In: *International Conference on Service-Oriented Computing*. Springer. 2018, pp. 230–245.

[67] Hao Yin, Xu Zhang, Hongqiang H Liu, Yan Luo, Chen Tian, Shuoyao Zhao, and Feng Li. “Edge provisioning with flexible server placement”. In: *IEEE Transactions on Parallel and Distributed Systems* 28.4 (2016), pp. 1031–1045.

AUTHORS

Athanasios Tsipis received in 2015 his B.Sc. degree in Informatics from the Department of Informatics, Ionian University, Greece. In 2017 he obtained his M.Sc. degree in Computer Science and Information Systems, for which he was awarded a tuition-free scholarship, while in 2021 he received his Ph.D. degree in Informatics, both from the same Department. His research interests concentrate on multimedia cloud computing, cloud gaming, fog/edge computing, network optimization, performance issues in IoT and wireless networks, facility location allocation, service placement, etc. Since 2015 he has been an active member and researcher of the “Networks, Multimedia and Security Systems Laboratory” (NMSLab), participating in various EC research and local development projects. Currently, he serves as a teaching assistant at the Department of Digital Media and Communication, as well as an academic scholar at the Department of Informatics, at Ionian University. He has been a reviewer of numerous conferences and journals, and in 2021 he was the recipient of the best paper award from the 26th IEEE Symposium on Computers and Communications.

Konstantinos Oikonomou received his M.Eng. degree in Computer Engineering and Informatics from the University of Patras in 1998. In September 1999 he received his M.Sc. degree in Communication and Signal Processing from the Electrical Department, Imperial College (London). He received his Ph.D. degree in 2004 from the Informatics and Telecommunications Department, University of Athens, Greece.

His Ph.D. thesis focuses on medium access control policies in ad hoc networks. He has served as the Dean of the Faculty of Information Science and Informatics of the Ionian University, Corfu, Greece. Since 2006 he is a faculty member in Computer Networks, currently a Professor, at the Department of Informatics, at the same University. He has also served as the Head of the Department and Director of the Postgraduate Studies. Between December 1999 and January 2005 he was employed at Intracom S.A, as a research and development engineer. His research interests involve medium access control in ad hoc networks, performance issues in wireless networks, information dissemination, service discovery, facility location, energy consumption in wireless sensor networks, and network cost reduction in cloud computing environments. Professor K. Oikonomou has a long experience in wireless systems and has been coordinating a number of EC research projects in the computer networks area, and various local development projects (e.g., virtual worlds). He is currently a member of the editorial boards of Computer Networks Journal (Elsevier) and Journal on Future and Evolving Technologies (ITU). He has been a reviewer and TPC member of numerous conferences and journals, and he holds an award for the best paper from the Hawaii International Conference on System Service, as well as an award for the best paper from the 26th IEEE Symposium on Computers and Communications.