Received: 25 December 2016 | Revised: 11 December 2017 | Accepted: 25 January 2018
DOI: 10.1002/cpe.4454

WILEY

SPECIAL ISSUE PAPER

Optimal configuration of active and backup servers for augmented reality cooperative games

Armir Bujari | **Luigi De Giovanni** | **Claudio E. Palazzi**

Department of Mathematics, University of Padua, 35121 Padua, Italy

Correspondence
Armir Bujari, Department of Mathematics, University of Padua, 35121 Padua, Italy.
Email: abujari@math.unipd.it

Funding information
Università degli Studi di Padova, Grant/Award Number: PRAT CPDA137314 and PRAT CPDA151221

Summary

Interactive applications as online games and mobile devices have grown in popularity. From their combination, new and interesting cooperative services could be generated. For instance, gamers endowed with Augmented Reality (AR) visors, connected as wireless nodes in an ad-hoc network, can interact with each other while immersed in the game. To enable this vision, we discuss here a hybrid architecture enabling game play in ad-hoc mode instead of the traditional client-server setting. In our architecture, one of the player nodes also acts as the server of the game, whereas other backup server nodes are ready to become active servers in case of network disconnection, ie, due to low energy level of the currently active server. This allows to have a longer gaming session before a disconnection occurs due to energy exhaustion. In this context, the server election strategy with the aim of maximizing network lifetime is not so straightforward. To this end, we have analyzed this issue through a Mixed Integer Linear Programming (MILP) model considering static network topologies and both numerical and simulation-based analysis shows that the backup servers solution fulfills its design objective. Contemplating for mobility, we present a simulation study employing actual packet routing protocols and discuss the tradeoffs that emerge.

KEYWORDS
backup server, MANET, mixed integer linear programming, online game

1 | INTRODUCTION

In recent years, interactive applications such as online games and mobile devices have become more and more popular, generating a huge market. This success is also accompanied by tough technical challenges that have attracted the interest of researchers and practitioners all around the world. Indeed, although online games require only a little amount of bandwidth, their performance is subject to strict per-packet delay requirements and to scalability issues that can be solved only through specific solutions and architectures.[1-3]

The combination with wireless-enabled mobile devices fosters new gaming scenarios. For instance, think of a multiplayer game based on the user position and proximity, ie, a game that is played outside, on the street, or in courtyards, or close to cultural heritage sites.[4] Players may gather around a certain spot and start a gaming session on their mobile devices, exploiting the device features (eg, connectivity, GPS, gyroscope, etc), rendering the game play more attractive. These applications may be coupled with Augmented Reality (AR), or even with Virtual Reality (VR), through helmets and wireless connectivity so as to enable players to interact with the AR/VR world and with each other (eg, see Figure 1).[5-8]

In these envisioned scenarios, players will be engaged in an outdoor, connectivity-, and proximity-based game. However, in order to enable this scenario, we have to address at least three main issues. First, the main model for networked games is client-server, with the client run by the player's PC and the server located remotely in the Internet.[9-11] Clearly, we could have fully distributed games, with nodes interacting in a peer-to-peer (P2P) fashion. However, generally existing games do not work in this way; the client-server paradigm is the preferred one as it allows a central control of game access and keeps a unique game-state progression that avoids cheating and discrepancies among the game state as seen by each player.[12] Therefore, aiming to ease the adaptation of existing successful online games, without completely re-designing their software we need to project the client-server model on the scenario we are considering. Unfortunately, this is not simple since an ad-hoc network is generally not connected to the Internet. Therefore, conversely from existing online games for home users, one of the players' devices has also to act as the server.

Concurrency Computat Pract Exper. 2018;e4454.
https://doi.org/10.1002/cpe.4454

FIGURE 1 An example of AR gaming

Second, it is well known that intermittent links represent a major issue for mobile ad-hoc networks (MANETs). Indeed, mobility could move one or more nodes out of range of the original MANET, becoming unable to reach the game server node and continue with the interaction. Even worse, the server node itself may get out of range for the rest of the network, thus interrupting the game session for the whole MANET.

Third, energy consumption/limitation is certainly a major concern when considering applications based on the connectivity of portable devices. This limited energy can be quickly consumed by game-related computation, visualization, and communication. Clearly, the worst energy consumption is experienced by the node also acting as the server of the game; that device will experience a much faster decrease of its energy reserve as it will be a central node for the whole MANET. In fact, it will have to receive all game events from other players, compute game state updates, and transmit these updates back to the other players.

To this end, we consider the use of a hybrid architecture in which more than one node is able to act as server and, in turn, one of these nodes becomes the *active server*, whereas the others remain as *backup servers*.[13] In this approach, in addition to the activity described, with a certain periodicity, the active server forwards the whole game state view to the backup servers, so as to have them ready to take charge. Having more servers taking turns in becoming active, distributes the highest energy consumption burden on more nodes, thus allowing a longer gaming session. However, communication among nodes is responsible as well for energy consumption, and the amount of communication depends on routing strategies and on servers' location, which are hence crucial to maximize the time before the first node in the network exhausts its battery, thus interrupting the game.

In this context, we analyze the Client-Server MANET Configuration Problem (CSMCP) aiming at finding an optimal configuration of a generic MANET in terms of servers' location and routing strategy.[14] To this aim, we consider a given number of servers and a fair rotation in terms of being the active server. Communications among nodes are done in a classic, unicast way although communication may involve multi-path routing.

Our aim is to maximize the time before any interruption happens in the game (ie, any node runs out of energy), conversely, minimize the total amount of energy consumed. In this paper, we use an analytical model to solve the problem in a static network scenario and then validate the suggested solution through simulations performed employing the well-known freely available open-source NS-2 simulator. Furthermore, contemplating for device mobility, we present a simulation analysis employing actual packet routing protocols.

The rest of this paper is organized as follows. In Section 2, we provide a concise survey of interactive networked AR cooperative applications that can benefit from our proposal. Next, in Section 3 we sketch the idea behind the backup servers solution, outlining the issues and paving the roadmap toward their resolution. Next, in Section 4, we provide a model aimed at tackling the problem of backup servers placement in a static network scenario, and, through Section 5, we discuss both a numerical and a simulation analysis showing that our approach fulfills its design objective. Section 6 investigates the impact of node mobility while actual packet routing protocols. Considering the affinity among the ad-hoc and P2P networking paradigms, in Section 7, we provide a concise survey on P2P architectures in the online gaming domain. Finally, in Section 8, we draw the conclusions and point out future research directions.

2 | INTERACTIVE NETWORKED COOPERATIVE AR APPLICATIONS

Augmented Reality (AR), also referred to as Mixed Reality, is a technology that layers computer-generated enhancements (eg, images, links to web pages, and 3D objects) on top of the existing reality (ie, physical world).[8,15-17] It has a common origin with Virtual Reality (VR) but it differs as it immerses the user in a world with both physical objects and virtual ones.

In this context, cooperative AR applications have received a lot of attention since the great success of games such as Ingress and Pokémon Go. An increasing interest is coming also from researchers due to the technical challenges of AR applications. Indeed, cooperative AR is potentially able to enhance the way users perceive the world and interact with/through it. By overlaying digital data over the physical view, it is possible to provide users with a shared, synthetic, and information-based "sixth sense." Possible applications for this technology are limited only by our imagination.[18,19]

As representative examples, we provide in the following a concise overview of potential interactive networked cooperative AR applications whose deployment could benefit from our study.

Safety ensuring applications. From a safety perspective, AR could be exploited to provide richer information in scenarios with limited visibility (eg, under water, in outer space, and scenarios with adverse meteorological conditions). In this context, virtual lines and objects, even Head-Up Displays (HUDs), can be used to aid the navigation. An emergent scenario is that of disaster relief, ie, the support and coordination of first aid squads in an emergency area after a crisis (eg, earthquake, flooding, and major accident).[20,21] Facilitating rescue operations, first aid squads may utilize HUDs with superimposed information about dangers and people's health conditions while coordinating through voice communication.

Medical applications. Applications of AR in the medical context can be exploited to enhance a doctor's view of a patient, pre-treatment planning, non-invasive surgery, and remote operations.[22] We could envisage applications where data generated by magnetic resonance, computed tomography scans, X-rays, and ultrasounds are directly projected over the patient's body or over a remote manikin. This mirrored patient could enable the doctor to perform precise operations without the need for large incisions and wherever the doctor(s) and the patients are located with respect to each other.[23,24]

Cultural heritage applications. AR could be exploited to provide a rich immersive experience to museum visitors. Users could interact with (the digital representation of) a piece of art, choose the level of reconstruction of artifacts and historical sites, and, in general, foster new participative learning applications.[25-27] Furthermore, investigators can use digital notes superimposed on archaeological sites or paintings to attach information to the object of study in a non-invasive way and make it available to other researchers to facilitate research in collaboration.[28]

Maintenance and assembly. Complex machineries require extensive assembly and maintenance efforts. Alleviating these burdens, AR could be employed in an intelligent manner, projecting online instructions, drawings, step-by-step animated examples, known issues, and previously performed reparations over the operator's view of the machinery.[29] Furthermore, to help any operator that may be in front of the broken machinery, suggestions could be prepared in real-time by remote highly specialized operators and projected over the machinery, along with instructions and requests simultaneously exchanged by voice communication.

Annotations. Reminders and notes could be replaced by digital ones and placed in an AR environment.[30,31] As a major advantage, digital notes could be easily customized to be public or specifically destined to a certain user (and existing only in the AR environment as seen by this user); moreover, they could also be automatically generated from databases (eg, labels in a store) and instantaneously modified over the entire AR environment with just one click/event in a remote location.

Entertainment applications. The AR technology has been extensively used even in entertainment.[32-34] For instance, more and more magazines include the possibility to augment the printed pages with digital objects through the smartphone. Furthermore, games based on augmented reality have been a major success (eg, Ingress and Pokémon Go). Other applications include advertisement, surrounding discovery, and the possibility to improve how we enjoy events (eg, Google Goggles, Layar, and many other less known but fun apps that can be found by searching “AR” on any app store).

Cooperative AR gaming. A specific branch of entertainment applications are interactive games. In this context, applications are characterized by very tough requirements and technical challenges, making them a very interesting case study where AR technology could be coupled with game action messages in order to create an AR arena.[35] In this setting, players endowed with AR helmets could be engaged in teams against virtual enemies and/or against each other while immersed in a mixed-reality arena.[6,10]

3 | BACKUP SERVERS IN MANETS

The targeted application scenario is the multiplayer interactive gaming, played by humans operating wireless-enabled devices in an outdoor environment exploiting ad-hoc connectivity. Network devices could be smart phones, PDAs, tiny laptops, or other portable devices customized for this kind of application (eg, helmets for AR). It is noteworthy to point out that our solution is not limited to this context; the underlying requirements of our solution are common to a wide range of distributed applications.

Ad-hoc connectivity has the benefit of freeing the user from the needs of an Access Point (AP), hence facilitating the emergence of a gaming session whenever the necessary number of players are gathered in a certain location. This kind of connectivity could seem to be directly linked with having a P2P game architecture. Yet, the majority of current online games follow a client-server paradigm. Therefore, the best way forward is to adopt a client-server approach even in our ad-hoc network scenario by introducing a middleware, which transparently manages network configuration and resources.

A client-server architecture is a straightforward solution with a fixed infrastructure and dedicated game servers. Unfortunately, in the considered MANET scenario, this implies that one of the nodes must also act as server and, by doing so, it has the computational burden, which translates in more battery consumption. Indeed, the server node is subject to more in-out traffic flows when compared to any other client. Moreover, the server node needs to compute the updated game state, hence the player hosting the server will drain its battery more quickly than the other players. This will result in shorter gaming sessions. Once the server is out of energy, the game will be interrupted. In addition, due to mobility, the network graph might change over time, eg, splitting in two or more connected groups and having just the group of players still connected to the server-player able to continue the gaming session.

Addressing the issue, our solution adopts more than one server. At each moment in time, only one is active while the others are kept synchronized in the background.[36] An active server is periodically selected among all candidate servers according to some criteria (later on) whose objective is to distribute battery consumption over all servers. Note that a node that acts as a server is also a client and thus participating in the game.

Because of node movements, most of the time a node will not be able to reach all the other nodes but only a few ones. We can see each node as a member of a partition, ie, a subset of nodes that reach each other. A partition could change at any time, eg, a node could go out of its partition and enter another partition; or an isolated client can join a partition and begin to play. We note that we allow, at most, one active server in each partition.

Communications between nodes are made in unicast, like in the actual games; in particular, when the active server has to send a packet to other nodes, it sends a copy for each destination, without relying on broadcast or multicast. Clearly, broadcast/multicast communication can be added later; their integration with our solution can generate even better performance. For the sake of clarity, now, we summarize node types in the simulation:

- **active server**: a node that plays and holds the game status, communicates with all nodes in its partition and keeps synchronized backup servers;
- **backup server**: a server that plays and can be chosen as active server in the future;
- **client**: a node that only plays the game.

In the following section, we introduce our proposed solution based on a Mixed Integer Linear Programming (MILP) formulation, discussing how it tackles the issues.

4 | A MIXED INTEGER PROGRAMMING FORMULATION

The formulation of the CSMCP in the static case depends upon the following input parameters: the number of servers L, the set N of the network nodes, and the set E of communication links. Links correspond to ordered pairs $(i, j) \in N \times N$, and each link (i, j) has a maximum bandwidth U_{ij}. It goes without saying that link (i, j) exists when nodes i and j are in communication range with each other. Regarding the energy resource, each node v is equipped with a battery with an initial energy level (A_v). Energy consumption is modeled by considering the average energy consumed per time unit by v when acting as a client (P_v^C) and the additional energy consumed by v in case it acts as active (respectively, backup) server (P_v^S, respectively, P_v^D). The active server is selected among the candidate servers in a round robin fashion, thus each server will be active during a time interval of fixed size, which is periodically repeated.. The time in which a server is active is called *shift* and we have L different shifts. It is noteworthy to point out that the model could embody more complex policies (eg, a weighted round robin) by defining a number of rounds larger than the server pool cardinality, attributing to more capable node a higher number of rounds.

For a given node v, three types of data packets are modeled, ie, packets to be routed from v acting as a client to the active server (type C), packets from the active server to v acting as a client (type S), and packets from the active server to v acting as a backup server (type D). Data packets have different sizes and transmission rates, depending on their type and on the node v. In specific, B_v^C, B_v^S, and B_v^D are the given bandwidth values required for packets of type C, S, and D related to v.

Contemplating for energy consumption while transmitting and receiving data, the model employs T_{kjv}^{τ} and R_{ikv}^{τ}, referring to the energy per time unit consumed by node k for transmitting (respectively, receiving) on link (k, j) (respectively, (i, k)) the packets of type $\tau \in \{C, S, D\}$ related to node v.

Based on the aforementioned notation, one MILP model for CSMCP is the following[14]:

$$\min \ \alpha + \epsilon \sum_{v \in N} \zeta_v \tag{1}$$

$$\text{s.t. } A_v \ \alpha \geq \zeta_v, \quad \forall \ v \in N \tag{2}$$

$$P_v^C + \left(\frac{1}{L} P_v^S + \frac{L-1}{L} P_v^D \right) \sum_{l=1}^{L} \sigma_v^l + \frac{1}{L} \sum_{l=1}^{L} \left(\gamma_{lv}^T + \gamma_{lv}^R \right) = \zeta_v, \quad \forall \ v \in N \tag{3}$$

$$\sum_{v \in N} \sigma_v^l = 1, \quad \forall \ l = 1..L \tag{4}$$

$$\sum_{l=1}^{L} \sigma_v^l \leq 1, \quad \forall \ v \in N \tag{5}$$

$$\sum_{(k,j) \in \bar{E}} x_{kj}^{lv} - \sum_{(i,k) \in \bar{E}} x_{ik}^{lv} = \begin{cases} 1, & \text{if } k = v \\ -1, & \text{if } k = \bar{s}, \\ 0, & \text{otherwise} \end{cases} \qquad \forall \ k \in \bar{N}, v \in N, l = 1..L \tag{6}$$

$$\sum_{(k,j)\in\bar{E}} y_{kj}^{lv} - \sum_{(i,k)\in\bar{E}} y_{ik}^{lv} = \begin{cases} -1, & \text{if } k = v \\ 1, & \text{if } k = \bar{s}, \\ 0, & \text{otherwise} \end{cases} \qquad \forall\, k \in \bar{N}, v \in N, l = 1..L \tag{7}$$

$$\sum_{(k,j)\in\bar{E}} z_{kj}^{lv} - \sum_{(i,k)\in\bar{E}} z_{ik}^{lv} = \begin{cases} -\sum_{r=1,r\neq l}^{L} \sigma_v^r, & \text{if } k = v \\ \sum_{r=1,r\neq l}^{L} \sigma_v^r, & \text{if } k = \bar{s}, \\ 0, & \text{otherwise} \end{cases} \qquad \forall\, k \in \bar{N}, v \in N, l = 1..L \tag{8}$$

$$x_{k\bar{s}}^{lv} \leq \sigma_k^l,\; x_{\bar{s}k}^{lv} \leq \sigma_k^l,\; y_{k\bar{s}}^{lv} \leq \sigma_k^l,\; y_{\bar{s}k}^{lv} \leq \sigma_k^l,\; z_{k\bar{s}}^{lv} \leq \sigma_k^l,\; z_{\bar{s}k}^{lv} \leq \sigma_k^l,\; \forall\, k, v \in N, l = 1..L \tag{9}$$

$$\sum_{v\in N} B_v^C x_{ij}^{lv} + B_v^S y_{ij}^{lv} + B_v^D z_{ij}^{lv} \leq U_{ij}, \quad \forall\, (i,j) \in E, l = 1..L \tag{10}$$

$$\sum_{v\in N} \sum_{(i,k)\in E} R_{ikv}^C x_{ik}^{lv} + R_{ikv}^S y_{ik}^{lv} + R_{ikv}^D z_{ik}^{lv} = \gamma_{lk}^R, \quad \forall\, k \in N, l = 1..L \tag{11}$$

$$\sum_{v\in N} \sum_{(k,j)\in E} T_{kjv}^C x_{kj}^{lv} + T_{kjv}^S y_{kj}^{lv} + T_{kjv}^D z_{kj}^{lv} = \gamma_{lk}^T, \quad \forall\, k \in N, l = 1..L \tag{12}$$

$$\sigma_j^l \leq 1 - \sigma_i^{l+1}, \quad \forall\, l = 1..L-1, i, j \in N : i < j \tag{13}$$

$$\alpha \in \mathbb{R}; \qquad \zeta_v \in \mathbb{R}_+, \quad \forall\, v \in N \tag{14}$$

$$\gamma_{lv}^T, \gamma_{lv}^R \in \mathbb{R}_+, \quad \forall\, v \in N, l = 1..L \tag{15}$$

$$\sigma_v^l \in \{0, 1\}, \quad \forall\, v \in N, l = 1..L \tag{16}$$

$$x_{ij}^{lv}, y_{ij}^{lv}, z_{ij}^{lv} \in [0, 1], \quad \forall v \in N, (i,j) \in \bar{E}, l = 1..L. \tag{17}$$

The model relies on the following decision variables: σ_v^l; Boolean variables taking value 1 if node v is the active server in the shift $l \in \{1..L\}$, 0 otherwise; and x, y, and z, determining the routing strategy. In particular, x_{ij}^{lv} (respectively, y_{ij}^{lv} and z_{ij}^{lv}) is the percentage of packets of type C (respectively, S and D) that, in shift l, are routed on link (i,j). Moreover, ζ_v denotes the average consumption per time unit of node v; γ_{lv}^T (respectively, γ_{lv}^R), the average consumption of node v for transmitting (resp. receiving) during the shift l; and α, the inverse of the game duration.

The objective function (1) minimizes a weighted sum of α and the total consumption. A small-enough parameter ϵ gives priority to α minimization (hence, game duration maximization).

Each node v exhausts its battery after $\frac{A_v}{\zeta_v}$ time units and the game duration is $\frac{1}{\alpha} = \min_{v\in N} \frac{A_v}{\zeta_v}$. In order to take the model linear in the variables ζ_v, we equivalently force, by constraints (2), $\alpha \geq \frac{\zeta_v}{A_v}$ for all the nodes and, thanks to the objective function, the optimal solution will have α equal to the most tightening bound.

Constraints (3) set variables ζ_v by summing up the energy per time unit consumed by node v. The first addend is the energy consumed as a client (we recall that a node is always acting as a client). The second addend gives the consumption as an active or backup server, ie, this component is only considered if v is chosen to be a server (ie, $\sum_{l=1}^{L} \sigma_v^l = 1$); different weights are associated to P_v^S and P_v^D since a server is active during one shift, ie, a fraction $\frac{1}{L}$ of the game duration, and backup during the remaining $\frac{L-1}{L}$ fraction. The last addend represents the consumption for communications as computed, for each shift, by the following constraints (11) and (12), taking into account that each shift takes $\frac{1}{L}$ of the game duration.

By constraints (4), we choose exactly one node as the active server for each shift. Moreover, a node will act as server in at most one shift, thanks to (5).

Ensuring the multi-path routing of packets are the multi-commodity flow conservation constraints (6), (7), and (8). Each constraint (6) considers the packets generated by node v as a client (type C) during shift l. For any node k, the left-hand side is the difference between the percentage of these packets leaving k through its outgoing arcs (k,j), and the percentage entering k through its incoming arcs (i,k). This difference has to be 1 if k is the same node v (all the packets must leave v), -1 if k is the active server in l (all the packets reach the active server), and 0 otherwise (the other nodes act as transit nodes or are not involved at all). The condition for k to be the active server depends on variable σ and cannot be directly stated in a linear formulation. As a consequence, a dummy node $\bar{s}$ is introduced, together with dummy links $(\bar{s}, v)$ and $(v, \bar{s})$, for all $v \in N$. $\bar{N}$ and $\bar{E}$ denote the sets of nodes and links extended to dummy elements, and flow variables x (as well as y and z) are extended to $\bar{E}$. Constraints (6) set the difference between the outgoing and incoming flow to -1 for the dummy node $\bar{s}$, which collects the flow directed to the active server. In order to force this flow to reach the real active server, constraints (9) activate the dummy links between $\bar{s}$ and a node i during the shift l, only if i is configured as the active server for the same shift l. As a consequence, all the traffic to the dummy server $\bar{s}$ must flow through the actual selected server. Hence, variables x give feasible routings for the packets of type *C*, differentiated by shifts. Similarly, constraints (7) ensure that *y* give feasible routings for the packets of type *S*. In this case, node $\bar{s}$ will generate all the type *S* traffic, which, by the activating constraints (9), will reach and leave the chosen real server first, and then, it will be routed to the client *v*. Type *D* traffic is routed by constraints (8). Notice that the traffic is generated at $\bar{s}$, and hence received by the actual active server first, and by node *v* after, only if *v* acts as server in a different shift (in this case, the value of the sum in the right-hand side is 1); if *v* is not a server (or it is the shift *l* active server), no type *D* traffic is generated for *v*.

The routing variables give, for each link and each shift, the percentage of packets directed to or generated at any node for the three types of traffic. Constraints (10) sum up all the bandwidth values required on each link in a specific shift (left-hand side) and bound the total value by the link capacity.

Constraints (11) compute the value of variables γ^R. For each node *k* and shift *l*, they sum up the energy consumed to receive packets on all the incoming arcs, with reference to the three traffic types and to all the transmitting or receiving nodes. Similarly, constraints (12) compute the value of variables γ^T (consumptions due to transmissions).

We notice that the game duration and the total consumption does not depend on the order in which the selected servers are activated, whereas, from the model perspective, different orders correspond to different solutions (different values for variables σ), even if they are equivalent in terms of the objective function. In other words, the set of feasible solutions shows symmetries related to equivalent permutations of shift indexes, which may impact the efficiency of the solution process. Some of this symmetries are broken by the technical constraints (13). They use any a priori order $\prec$ between nodes to exclude some permutations while guaranteeing that at least one equivalent solution remains feasible.

5 | MODEL EVALUATION

Preliminary tests have been conducted, and a representative subset of the results is shown in the following. Results refer to a prototype implementation of the proposed model in OPL[37] running on an Intel Core i7 2.2 GHz CPU with 8 GB RAM and using, as optimization engine, Cplex 12.4[37] with default settings.

To validate the outcome, we have utilized the well-known freely-available open-source NS-2 simulator (ver. 2.33).[38] Each configuration of the simulation has been run 40 times and the outcome has been averaged. In our experiments, we have considered the aforementioned various topologies, with 3 servers and 9 KJ of energy initially present in every node in the MANET. The active and backup servers in each configuration are chosen according to the solution provided by solving the corresponding MILP model.

In Table 1, we present a comparison between model and NS-2 outcomes when 3 servers (1 active server and 2 backup ones, taking turns) are used. Each row reports the name of the instance in the first column (see Figure 2). The second column reminds the total number of nodes present in that specific topology. The third and fourth columns represent the total amount of time a game session lasts before any node runs out of energy, considering the proposed model and the simulation outcome, respectively. The last column is the standard deviation corresponding to the obtained simulation time value. There is a clear correspondence between model values and simulation ones.

TABLE 1 Model vs. simulations: game session time

		Model	NS-2	
Topology	Nodes	Time (s)	Time (s)	Std dev
btn25	25	4122.6	3953.1	53.5
btn41	41	2466.6	2442.7	83.3
btn48	48	4038.0	3991.3	41.5
9×3	27	5392.8	4936.2	78.4
12×4	48	4168.8	4013.0	15.0
7×7	49	4122.6	4019.0	19.7
8×8	64	3531.0	3601.1	71.2
10×10	100	2626.8	2714.6	93.6

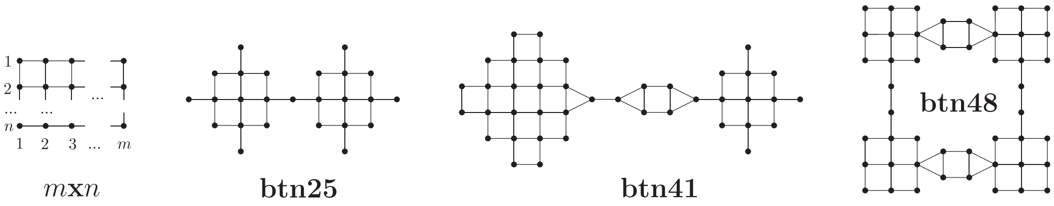

FIGURE 2 Sample topologies

TABLE 2 Model vs. simulations: energy consumption

		Model		NS-2		
Topology	Nodes	Energy (J)	Avg Energy (J)	Energy (J)	Avg Energy (J)	Std dev
btn25	25	147011.9	5880.5	171101.3	6844.1	133.7
btn41	41	170787.4	4165.5	233484.4	5694.7	99.4
btn48	48	315367.8	6570.2	353319.9	7360.8	72.4
9×3	27	209887.8	7773.6	224102.4	8300.1	141.3
12×4	48	318079.4	6626.7	369512.9	7698.2	80.4
7×7	49	301650.6	6156.1	360175.3	7350.5	70.7
8×8	64	355889.5	5560.8	438137.8	6845.9	50.4
10×10	100	458534.2	4585.3	603112.1	6031.1	37.9

TABLE 3 1 vs. 3 servers: game session time

Topology	1 Server Time (s)	3 Servers Time (s)	Increment (%)
btn25	3142.5	3953.1	25.9
btn41	2074.2	2442.7	17.8
btn48	3023.0	3991.3	32.0
9×3	4108.9	4936.2	20.1
12×4	2916.7	4013.0	37.6
7×7	2903.6	4019.0	38.4
8×8	2740.2	3601.0	31.4
10×10	2241.5	2714.6	21.1

TABLE 4 1 vs. 3 servers: energy consumption

Topology	1 Server Energy (J)	3 Servers Energy (J)	Increment (%)
btn25	149407.3	171101.3	14.5
btn41	207026.4	233484.4	12.9
btn48	298689.6	353319.9	18.3
9×3	196357.1	224102.4	14.1
12×4	309215.8	369512.9	19.5
7×7	294044.6	360175.3	22.5
8×8	371712.7	438137.8	17.9
10×10	537293.6	603112.1	12.3

In Table 2, we present the different energy consumption reported by our model and by NS-2 for the different considered topologies. In particular, Energy (J) represents the total amount of energy in Joules cumulatively consumed by all the nodes before any node runs out of energy. Instead, Avg Energy (J) is simply Energy (J) divided by the number of nodes thus representing the average consumption by each node. As this value is obtained averaging the outcome of multiple simulation runs, for NS-2, we also report the standard deviation. Again, there is a clear correspondence between model values and simulation ones.

For completeness, we also report a comparison between a classic scenario with only one server (located in the middle of the MANET) and our solution with 3 servers taking turns in being active. To this aim, Table 3 compares the game session time for the two cases. As it is evident, with 3 servers, the duration of the game is extended considerably before any node is out of energy (interrupting the game session). Obviously, this coupled with an increment of the total energy cumulatively consumed in the MANET (see Table 4); yet, this is acceptable since we succeed in incrementing the amount of continuous gameplay with no disconnections and since the increment of the energy cumulatively consumed is even smaller than the extended duration of the game session.

In summary, these results point out that the proposed model for the CSMCP can be effectively used to optimally solve small instances of the problem. Furthermore, from our analysis, it emerges that the servers should be fairly distributed around the center of the network (ie, the node chosen for $L = 1$). In fact, if all servers are too close to this center and, hence, to each other, when they are not active, they would consume energy for both backup-server duties and for message forwarding.

6 | MOBILITY

The evaluation discussed so far considers only static scenarios, with different topologies. This makes sense for two main reasons. First, we are aimed at understanding the reliability of our model in identifying efficient packet routes given a certain topology; we have hence to remove other possible factors that may hide the real properties of our model. Second, in a real world scenario, the considered nodes will move at a much slower speed than packet transmission and route computation. We could hence consider the topology evolution as a sequence of static states where our model is applicable.

Nonetheless, for the sake of completeness, we have considered some representative case studies among those analyzed in the previous section, and we have added mobility to the simulations. In particular, we compare a static case with no mobility against Reference Poing Ground Mobility (RPGM) and totally random mobility. RPGM is a particularly interesting case study as it represents the mobility of group of nodes moving independently but attracted by different points on the map at different times (eg, teams of players that move toward a point-of-interest or an enemy). We have combined NS-2 with the well known BonnMotion tool to generate mobility.[39] In particular, we have evenly (and randomly) divided the nodes into two groups (ie, two teams), and we randomly place a new couple of attraction points on the map every 3 minutes; one third of all the players move toward an attraction point, one third toward the other one, and one third keeps moving randomly.

In the aforementioned scenarios, we have compared three routing protocols, ie, AODV,[40] OLSR,[41] and OLSR-EE.[42] The first one is the classic choice when considering routing protocols in MANETs, and, for this reason, we cannot leave it out of our comparison. Instead, the second and (even more) the latter are well-known routing protocols that also take energy consumption and availability into consideration when determining packet routing; we have hence chosen them given the focus of our work.

We have considered two possible topologies for the nodes at the beginning of each simulation, ie, 7×7 and btn48. They embody representative cases of those considered in the model analysis and have almost the same number of nodes, 49 and 48, respectively. In both cases, we have considered the possibility to have just 1 server or 3.

In summary, Figures 3 and 4 show the game session time and the total energy consumption for the 7×7 starting topology when employing different routing protocols and mobility, whereas in Figures 5 and 6, we have the corresponding values for the btn48 starting topology.

From the charts, we can see that there is not a great difference in considering the 7×7 starting topology or the btn48 one. This is especially true when employing the RPGM or the random mobility model as the starting topology is quickly modified into something unrelated to the starting one and depending on the mobility of nodes.

In all the considered configurations, the case with 3 servers achieves better values than with just 1 server. This confirms the results we have obtained through the model and discussed in the previous section.

The mobility configurations slightly modify the performance. In particular, when employing random mobility, the protocols encounter some difficulties in keeping the routes active, thus achieving the lowest performance. Instead, RPGM performes even slightly better than the static one as

FIGURE 3 Game session time for the 7×7 starting topology

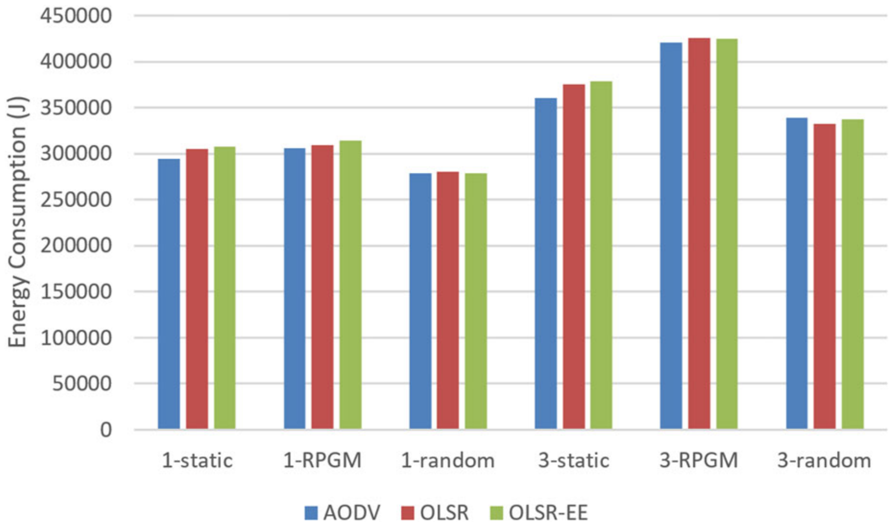

FIGURE 4 Energy consumption for the 7×7 starting topology

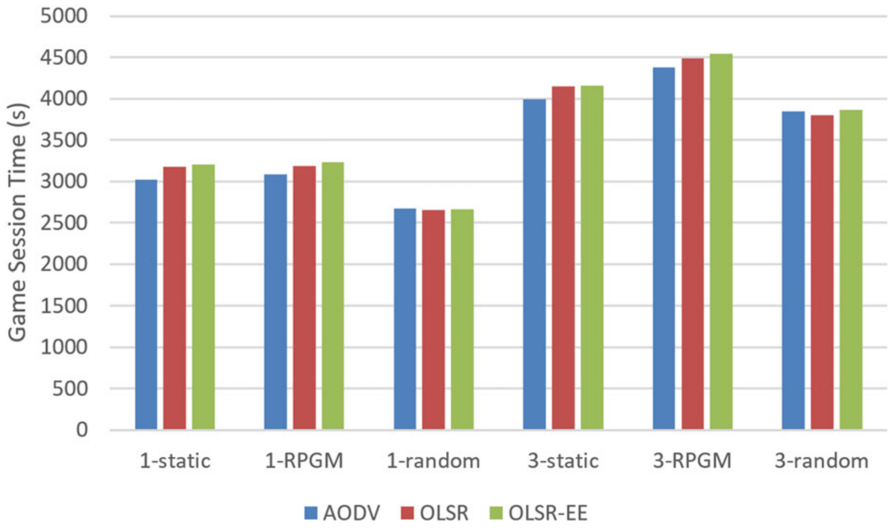

FIGURE 5 Game session time for the btn48 starting topology

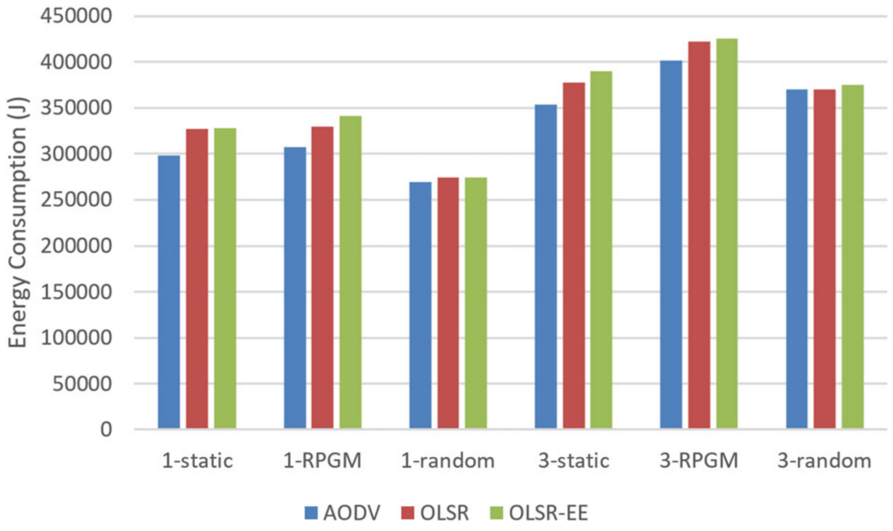

FIGURE 6 Energy consumption for the btn48 starting topology

this mobility model tends to group various nodes in direct transmission range of each other, thus diminishing the average number of hops needed to cover all nodes.

7 | RELATED WORK

The ad-hoc and peer-to-peer networking paradigms are strictly related as they both rely on direct communications amongst peering nodes. In the context of online gaming, P2P architectures represent an attractive alternative due to their decentralized and low cost nature. Indeed, nodes act both as client and servers distributing load and computational burden amongst each other.[43] To this end, a vast research effort has been dedicated culminating with a variety of proposals, which can be grouped into two categories, ie, (i) structured and (ii) unstructured P2P overlay approaches.

P2P structured overlay imposes a logical organization amongst nodes and rely on variations of the Distributed Hash Table (DHT) concept for game state distribution and update dissemination. In architectures following a structured approach, each node is assigned the responsibility of managing a primary copy of an object of the game world being this at the granularity of the single item or or an entire region of the game world. Depending on the proposal, redundancy is employed replicating objects in secondary destinations making the architecture more resilient against node departures, however, at a cost of a more expensive state update mechanism.[44]

On the other side, P2P unstructured overlay approaches impose no logical organization amongst nodes, and network connections are established randomly or through a probabilistic measure of affinity amongst nodes. Generally, these systems are based on a mutual notification scheme whereby neighboring nodes notify each other of game state changes on their respective Area of Interest (AoI), eg, line of sight.[45] In this context, gossiping protocols are employed for neighborhood discovery and network bootstrap.

Several systems combine both the approaches by employing more capable, coordinator, and nodes, which have the responsibility of managing a region(s) of the game world. Indeed, the game world is managed by employing a structured overlay organization amongst coordinator nodes and update dissemination is handled accordingly. At the same time, coordinator nodes exchange neighboring information amongst each other through gossiping and the exchange is done hierarchically involving only neighboring coordinators.

Despite their attractiveness, the factual deployment of P2P architecture approaches in the gaming context is limited to a few niche scenarios.[43] A major issue in this context is security; cheating is easier in P2P environments.[46] In addition, as argued, state in P2P approaches is distributed, hence requiring complex and difficult to achieve management and control functions for game state progression.

8 | CONCLUSION

While preserving the general aspects of our study and without loss of generality, we have envisioned a scenario where cooperative network AR games could be played locally whenever and wherever players engage. This flexibility comes with some technical challenges deriving from the contrasting nature of both technologies. Indeed, MANETs are inherently decentralized and potentially mobile while AR games are typically provisioned following the classical client-server paradigm.

Addressing the issues, we formulated a solution to the CSMCP in static network topology, validating our approach both through numerical and simulation analysis showing that it fulfills its design objective. A MILP formulation was provided and preliminary results point out that the proposed model can be effectively used to solve small instances to optimality. A complementary simulation study backs the findings, showing that network lifetime is indeed increased. In addition, addressing the scenario limitations, we present a simulation analysis in a real scenario comprised of mobile nodes employing actual routing protocols.

Of course, heuristics could be employed to solve larger instances of the problem, and convergence times could be reduced by exploiting additional information on instance settings (eg, on node battery) and suggestions emerging from the analysis of optimal servers' location. The outcome of this analysis could be exploited accordingly in a decentralized network setting. From an operational viewpoint, a network bootstrap period could be provisioned whereby nodes through gossiping exchange topological information, and this information is then exploited autonomously to elect the candidate server nodes.

In addition, the current problem formulation models only unicast communications among nodes; multicast/broadcast communication among server nodes brings additional benefits in terms of the amount of exchanged messages. This communication model fits well with an hierarchical network organization as in hybrid P2P overlays with server nodes intelligently managing and sharing game state among nodes.[12] These subjects are an object of ongoing research.

ACKNOWLEDGMENT

Our deepest gratitude goes to Silvia Cibola and to Alberto Giaretta for their technical help with CPLEX and NS-2. This work has been partially funded by the Università degli Studi di Padova through the projects PRAT CPDA137314 and PRAT CPDA151221.

ORCID

Armir Bujari http://orcid.org/0000-0002-1955-7699

REFERENCES

1. Kavalionak H, Carlini E, Ricci L, Montresor A, Coppola M. Integrating peer-to-peer and cloud computing for massively multiuser online games. *Peer Peer Network Appl*. 2015;8(2):301-319.
2. Maggiorini D, Ripamonti LA, Cappellini G. About game engines and their future. Paper presented at: 2016 International Internet of Things Summit; 2016; Rome, Italy.
3. Roccetti M, Ferretti S, Palazzi CE. The brave new world of multiplayer online games: Synchronization issues with smart solutions. Paper presented at: 2008 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing (ISORC); 2008; Orlando, FL.
4. Roccetti M, Marfia G, Semeraro A. Playing into the wild: a gesture-based interface for gaming in public spaces. *J Vis Commun Image Represent*. 2012;23(3):426-440.
5. Singhal S, Zyda M. *Networked Virtual Environments: Design and Implementation*. New York, NY: ACM Press/Addison-Wesley Publishing Co; 1999.
6. Cheok AD, Goh KH, Liu W, et al. Human Pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. *Pers Ubiquit Comput*. 2004;8(2):71-81.
7. Bujari A, Ciman M, Gaggi O, Marfia G, Palazzi CE. Paths: Enhancing geographical maps with environmental sensed data. Paper presented at: Proceedings of the 2015 Workshop on Pervasive Wireless Healthcare; 2015; Hangzhou, China.
8. Schmalstieg D, Hollerer T. *Augmented Reality: Principles and Practice*. Boston, MA: Addison-Wesley; 2016.
9. Cronin E, Kurc AR, Filstrup B, Jamin S. An efficient synchronization mechanism for mirrored game architectures. *Multimed Tools Appl*. 2004;23(1):7-30.
10. Ploss A, Wichmann S, Glinka F, Gorlatch S. From a single- to multi-server online game: A Quake 3 case study using RTF. Paper presented at: Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology; 2008; Yokohama, Japan.
11. Wright S, Tischer S. Architectural considerations in online game services over DSL networks. Paper presented at: 2004 IEEE International Conference on Communications; 2004; Paris, France.
12. Gerla M, Maggiorini D, Palazzi CE, Bujari A. A survey on interactive games over mobile networks. *Multimed Tools Appl*. 2013;13(3):212-229.
13. Palazzi CE, Kaiser A, Achir N, Boussetta K. A preliminary evaluation of backup servers for longer gaming sessions in MANETs. Paper presented at: Proceedings of the 3rd ICST International Conference on Simulation Tools and Techniques; 2010; Malaga, Spain.
14. De Giovanni L, Palazzi CE. Optimal client-server configuration of mobile ad-hoc networks. *Electron Notes Discrete Math*. 2013;41:495-502.
15. Benford S, Crabtree A, Flintham M, et al. Can you see me now? *ACM Trans Comput Hum Interact*. 2006;13(1):100-133.
16. Billinghurst M, Clark A, Lee G. A survey of augmented reality. *Found Trends Hum Comput Interact*. 2015;8(2-3):73-272.
17. Bulman J, Crabtree B, Gower A, Oldroyd A, Lawson M, Sutton J. Mixed reality applications in urban environments. *BT Technol J*. 2004;22(3):84-94.
18. Furini M. Mobile games: What to expect in the near future. Paper presented at: 2007 Proceedings of GAMEON; 2007; Duesseldorf, Germany.
19. Cacciaguerra S, Mirri S, Pracucci M, Salomoni P. Wandering about the city, multi-playing a game. Paper presented at: Proceedings of the 3rd IEEE Consumer Communications and Networking Conference; 2006; Las Vegas, NV.
20. Vassell M, Apperson O, Calyam P, Gillis J, Ahmad S. Intelligent dashboard for augmented reality based incident command response co-ordination. Paper presented at: Proceedings of the 13th IEEE Annual Consumer Communications and Networking Conference; 2016; Las Vegas, NV.
21. Tsai MK, Yau NJ. Improving information access for emergency response in disasters. *Nat Hazard*. 2013;66(2):343-354.
22. Indhumathi C, Chen W, Cai Y. Multi-modal VR for medical simulation. *Int J Virtual Reality*. 2004;8(1):1-7.
23. Windyga PS, Medina R, Onik GM. Augmented vision for minimally invasive abdominal cancer surgery. Paper presented at: Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society; 2003; Cancun,Mexico.
24. Glossop N, Wedlake C, Moore J, Peters T, Wang Z. Laser projection augmented reality system for computer assisted surgery. Paper presented at: 2003 International Conference on Medical Image Computing and Computer-Assisted Intervention; 2003; Quebec, Canada.
25. Wojciechowski R, Walczak K, White M, Cellary W. Building virtual and augmented reality museum exhibitions. Paper presented at: Proceedings of the 9th International Conference on 3D Web Technology; 2004; Monterey, CA.
26. Mirri S, Salomoni P, Pizzinelli A, Roccetti M, Palazzi CE. “di Piazza in Piazza”: Reimagining cultural specific interactions for people-centered exhibitions. Paper presented at: 2016 International Conference on Computing, Networking and Communications (ICNC); 2016; Kauai, Hawaii.
27. Bujari A, Ciman M, Gaggi O, Palazzi CE. Using gamification to discover cultural heritage locations from geo-tagged photos. *Pers Ubiquit Comput*. 2016;21(2):235-252.
28. Mazzeo R, Palazzi CE, Roccetti M, Sciutto G. Computer-assisted pigment identification in artworks. Paper presented at: Proceedings of the 2007 European Internet and Multimedia Systems and Applications; 2007; Chamonix, France.
29. Feiner S, Macintyre B, Seligmann D. Knowledge-based Augmented Reality. *Commun ACM*. 1993;36(7):53-62.
30. Feiner S, MacIntyre B, Haupt M, Solomon E. Windows on the world: 2D windows for 3D augmented reality. Paper presented at: Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology; 1993; Atlanta, GA.
31. Rekimoto J, Nagao K. The world through the computer: Computer augmented interaction with real world environments. Paper presented at: Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology; 1995; Pittsburgh, PA.
32. Layar application. https://www.layar.com/. Accessed September 2017.
33. Furini M. Fast play: a novel feature for digital consumer video devices. *IEEE Trans Consum Electron*. 2008;54(2):513-520.

34. Bartneck C, Hu J, Salem B, Cristescu R, Rauterberg M. Applying virtual and augmented reality in cultural computing. *Int J Virtual Reality*. 2008;7(2):11-18.
35. Maggiorini D, Quadri C, Ripamonti LA. Opportunistic mobile games using public transportation systems: a deployability study. *Multimedia Systems*. 2014;20(5):545-562.
36. Palazzi C, Bujari A. Evaluating outdoor games with the NS-2 and NS-3 simulation frameworks. Paper presented at: Proceedings of the 3rd IEEE Consumer Communications and Networking Conference; 2014; Las Vegas, NV.
37. IBM. ILOG CPLEX optimization studio. http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio. Accessed July 2016.
38. Network simulator 2 (NS-2). http://www.isi.edu/nsnam/ns/. Accessed September 2017.
39. BonnMotion - a mobility scenario generation and analysis tool. http://sys.cs.uos.de/bonnmotion/. Accessed September 2017.
40. Perkins C, Belding-Royer E, Das S. Ad hoc on-demand distance vector (AODV) routing. RFC 3561 Internet Engineering Task Force. 2003.
41. Clausen T, Jacquet P. Optimized link state routing protocol (OLSR). RFC 3626 Internet Engineering Task Force. 2003.
42. Kunz T. Energy-efficient variations of OLSR. Paper presented at: 2008 International Wireless Communications and Mobile Computing Conference; 2008; Crete Island, Greece.
43. Yahyavi A, Kemme B. Peer-to-peer architectures for massively multiplayer online games: a Survey. *ACM Comput Surv*. 2013;46(1):9.
44. Krause S. A case for mutual notification: A survey of P2P protocols for massively multiplayer online games. Paper presented at: Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games; 2008; Worcester, MA.
45. Chu YH, Rao SG, Seshan S, Zhang H. A case for end system multicast. *IEEE J Selected Areas Comm*. 2002;20(8):1456-1471.
46. Goodman J, Verbrugge C. A peer auditing scheme for cheat elimination in MMOGs. Paper presented at: Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games2008; Worcester, MA.

How to cite this article: Bujari A, De Giovanni L, Palazzi CE. Optimal configuration of active and backup servers for augmented reality cooperative games. *Concurrency Computat Pract Exper*. 2018;e4454. https://doi.org/10.1002/cpe.4454