International Journal of Wireless Communication Networks and Mobile Computing
Journal homepage: https://trpub.org/journal.php?id=GciKJ4WZtI

TRPub
IJWCNMC
International Journal of Wireless Communication Networks and Mobile Computing
Vol. 1 Issue 1 2026

Methodology Article **Open Access**

Energy-Aware Task Offloading for Mobile Edge Computing in Dense Urban Cells: A Proposed Modelling and Evaluation Framework

Nikola Jarrett[1*], Jack Ortega[2], Mateo Desai[3]
[1]Department of Electrical Electronics, Ebro Graduate Institute of Electrical Electronics, Thailand
[2]Department of Electrical Electronics, Periyar Academy of Electrical Electronics, Croatia
[3]Department of Electrical Electronics, Hejaz Institute of Electrical Electronics, Nepal

Article Info

Volume 1, Issue 1, 2026
doi: 10.62715/ijwcnmc.2026.v1.i1.pp15-20
Received: 14 May 2026
Accepted: 11 July 2026
Published: 15 September 2026

Abstract

Offloading a computation from a handset to a server at the network edge can save battery energy, but only when the radio energy spent on sending the task is smaller than the processor energy avoided. In dense urban deployments that balance is hard to judge. Small cells overlap, uplink interference couples the decisions of neighbouring users, edge servers attached to small cells have limited capacity, and the energy drawn by the infrastructure is seldom counted alongside that of the device. Published studies handle these matters under assumptions so varied that their conclusions are difficult to compare. This article proposes a seven-step framework for modelling and evaluating energy-aware offloading in such settings. The steps are to fix the scenario and the boundary of the energy account, describe the tasks, write explicit energy and delay terms for device and network, formulate the decision problem, choose a solution method suited to the timescale of the decision, compare it against stated baselines, and test the sensitivity of the conclusions. Each step is explained with the modelling choices it forces and the information it should record. The framework is assembled from established models in the literature. It is a proposal and has not been applied or validated here.

Keywords: *Mobile edge computing, Task offloading, Energy efficiency, Small cells, Resource allocation, Evaluation methodology*

© 2026 Nikola Jarrett et al. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

1. Introduction

The idea that a mobile device might extend its battery life by sending work elsewhere is older than edge computing. Kumar and Lu (2010) stated the condition in its simplest form: offloading saves energy when the computation is heavy and the data to be transferred are light, relative to the bandwidth available. The MAUI system showed that the decision could be taken per method call at run time on a real handset (Cuervo *et al.*, 2010), and the cloudlet proposal moved the server from a distant data centre to within one wireless hop so that delay would stop dominating (Satyanarayanan *et al.*, 2009). Multi-access edge computing, standardised by ETSI, places such servers inside the radio access network (ETSI, 2019).

* *Corresponding author: Nikola Jarrett, Department of Electrical Electronics, Ebro Graduate Institute of Electrical Electronics, Thailand. E-mail: nikola.jarrett@ebro.edu*

© 2026 Nikola Jarrett et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A large research literature has since treated offloading as an optimisation problem, and two surveys organise it well (Mach and Becvar, 2017; Mao *et al.*, 2017). Reading across that literature, however, one finds that the word energy covers different things. Some papers minimise the battery drain of one user, some the sum over users, some a weighted mix of energy and delay, and few include the base station or the server. The dense urban setting, in which the radio and computing resources of many small cells are shared among many users, makes these differences matter, because a decision that is frugal for one handset can be costly for its neighbours or for the network.

This article proposes a framework for setting up and evaluating an energy-aware offloading study in dense urban cells so that its assumptions are explicit and its results can be compared with others. It is a methodological proposal built from published models. It reports no simulation or measurement, and the framework has not been validated here.

2. Background

Three bodies of work supply the building blocks: models of the energy a device spends computing locally, models of the energy it spends transmitting, and formulations that couple many users through shared resources.

2.1. Local computing and transmission energy

For a processor built in complementary metal-oxide-semiconductor technology, dynamic power grows roughly with the cube of clock frequency when supply voltage is scaled with frequency, so the energy per cycle grows with the square (Burd and Brodersen, 1996). Offloading studies use this in the form $E = \kappa C f^2$, where C is the number of cycles the task needs, f the frequency and κ a chip-dependent coefficient. The consequence is that a device able to scale its frequency should run a task as slowly as the deadline allows. On the radio side, the energy to upload a task is the transmit power, plus circuit power, multiplied by the time taken, and the time is the task size divided by the achievable rate. Rate is commonly modelled with the Shannon formula over the signal-to-interference-plus-noise ratio. You *et al.* (2017) derive from these two models an offloading priority for each user that depends on its channel gain and its local computing energy.

2.2. Coupling between users in dense cells

When cells are small and close together, the uplink transmission of an offloading user raises the interference seen by others on the same channel, in its own cell and in adjacent ones. Their rates fall, their uploads take longer, and their transmission energy rises. Chen *et al.* (2016) modelled this as a game among users choosing channels and showed that it admits a potential function and hence an equilibrium reachable by distributed updates. Sardellitti *et al.* (2015) optimised transmit precoders and server allocation jointly across several cells. Zhang *et al.* (2016) addressed heterogeneous networks with macro and small cells. Where each user may reach several servers, association becomes a further variable (Tran and Pompili, 2019).

2.3. Decisions over time

Tasks arrive continually and channels fade, so a one-shot optimum is only part of the picture. Lyapunov drift-plus-penalty methods turn long-term average objectives and constraints into a sequence of per-slot problems without requiring knowledge of the underlying statistics (Neely, 2010), and were applied to offloading with energy-harvesting devices by Mao *et al.* (2016). Deep reinforcement learning has been used to produce binary offloading decisions quickly enough to follow fading (Huang *et al.*, 2020). Each family of methods rests on different information and runs at a different timescale, which the framework below asks the analyst to state.

3. The Proposed Framework

The framework consists of seven steps, shown in sequence in Figure 1. They are ordered so that each fixes something the next one needs. Nothing in them is new taken singly. The contribution lies in requiring all

of them and in the record each should leave.

3.1. Step 1: Fix the scenario and the energy boundary

The analyst first describes the deployment: the layout and density of small cells, whether a macro layer is present, the carrier frequency and bandwidth, the frequency reuse pattern, and which sites host servers. A recognised channel model for urban micro cells should be named instead of an ad hoc path-loss exponent (3GPP, 2020). The same step draws the boundary of the energy account. Three choices are defensible: the device alone, the devices plus the edge servers, or the whole access network. The choice must be declared, because offloading by construction moves consumption from batteries to mains-powered equipment, and a study that counts only one side of the transfer will always favour it.

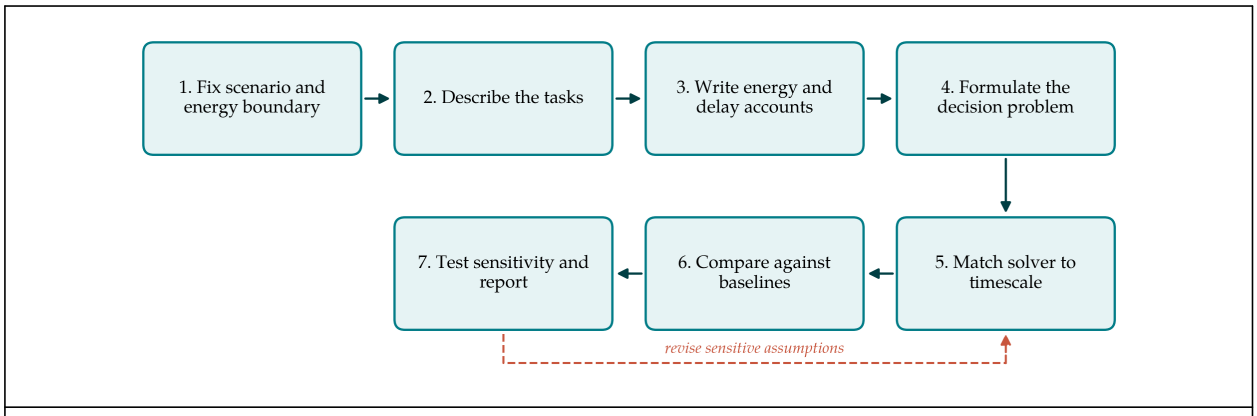


Figure 1: The seven steps of the proposed framework, with a return path from the sensitivity analysis

3.2. Step 2: Describe the tasks

Each task is characterised by its input size in bits, its workload in processor cycles, its deadline and the size of the result to be returned. The study should say whether tasks are atomic, so that the choice is binary, or divisible, so that a fraction can be offloaded, and how they arrive: one per user per slot, or as a random stream with queues. The ratio of cycles to bits deserves particular care, since it is the quantity on which the condition of Kumar and Lu (2010) turns, and the values assumed in published studies differ widely. Where no measured workload is available, the framework asks for results across a range of ratios and not at a single convenient value.

3.3. Step 3: Write the energy and delay accounts

Every term inside the boundary chosen in Step 1 is now written down. Table 1 lists the terms, the model usually adopted for each, and the omission most often encountered. Two deserve comment. A device that has offloaded a task does not switch off: it idles, with its radio in a connected state, until the result returns, and the energy of that wait can be comparable to the upload energy for long-running tasks. On the network side, measurements summarised in the widely used base-station power model show that small cells draw a large share of their peak power even when idle (Auer *et al.*, 2011), so the marginal network energy of one more offloaded task is small while the fixed cost of keeping an edge site awake is not.

3.4. Step 4: Formulate the decision problem

With the accounts in place, the decision variables are listed: for each user, whether or how much to offload, to which cell and server, on which sub-channel, at what transmit power, and with what share of server capacity. The objective is stated as a formula, and if it is a weighted sum of device and network energy or of energy and delay, the weights are given with a reason. Constraints cover deadlines, maximum transmit power, server capacity and, where relevant, backhaul. The analyst should also record the structure of the resulting problem. Joint binary offloading and resource allocation is in general a mixed-integer non-linear programme, and the literature handles it by decomposition into an allocation

sub-problem that is convex for fixed decisions and a combinatorial assignment solved heuristically (Tran and Pompili, 2019; Sardellitti *et al.*, 2015). Stating which part is solved exactly and which approximately is part of the formulation.

Table 1: Energy and delay terms that the proposed framework requires a study to state

Term	Usual model	Frequently omitted
Local computing energy	$\kappa C f^2$ with frequency scaling (Burd and Brodersen, 1996)	Limits on the range of frequency scaling
Uplink transmission energy	Transmit plus circuit power times task size over rate	Circuit power; interference from other cells
Device waiting energy	Idle power times remote execution and return time	Usually the whole term
Edge server energy	Idle power plus a load-dependent part	Idle power of servers kept on for few tasks
Radio access energy	Load-dependent base-station model (Auer *et al.*, 2011)	Backhaul and cooling
Delay	Upload, queueing, execution and download times	Queueing at the server; handover during a task

3.5. Step 5: Match the solution method to the timescale

Dense urban channels change within milliseconds, user positions within seconds and traffic patterns within minutes to hours. A method must be matched to the timescale at which its inputs are known. Convex optimisation with full channel knowledge suits slow, centralised decisions such as server placement or cell association. Game-theoretic updates suit distributed channel selection where users observe only their own interference (Chen *et al.*, 2016). Lyapunov methods suit per-slot control under long-term energy budgets (Mao *et al.*, 2016), and learned policies suit cases where the decision must be produced faster than an iterative solver allows (Huang *et al.*, 2020). The framework asks the analyst to state, for the chosen method, what information it assumes, who holds that information, how much signalling is needed to gather it, and how long a decision takes to compute. The energy of signalling and computation spent on the decision itself belongs in the account when it is not negligible.

3.6. Step 6: Compare against stated baselines

A proposed scheme means little without reference points. Four are required. Executing every task locally and offloading every task to the nearest cell bound the range of interest. A simple greedy rule, such as the priority ordering of You *et al.* (2017) applied per cell without coordination, shows what coordination adds. An exhaustive or branch-and-bound solution on instances small enough to solve shows how far the heuristic lies from the optimum. Evaluation is by simulation, preferably in a tool that models both radio and computing resources, of which EdgeCloudSim is one openly available example (Sonmez *et al.*, 2018). Results are reported as energy per completed task, broken down by the terms of Table 1, together with the share of tasks that miss their deadline and a measure of how unevenly energy savings are spread among users.

3.7. Step 7: Test sensitivity and report

The last step varies the assumptions that Steps 1 to 3 fixed: cell density, the cycles-to-bits ratio, circuit and idle power, server capacity and the energy boundary itself. A conclusion that reverses when the boundary moves from device to network is still a valid conclusion, but it must be reported as conditional. Where the outcome proves sensitive to an assumption that was chosen for convenience, the analyst returns to the relevant step, as the feedback path in Figure 1 indicates. The final report lists all parameter values, the random seeds and the number of independent runs, and makes the simulation code available where possible.

4. Fitting the Framework to Practice

The framework is meant to be usable by a research student planning a simulation study and by an engineer comparing vendor claims. For the first, Steps 1 to 4 amount to a checklist for the system-model section of a paper, and Steps 6 and 7 for its evaluation section. For the second, the energy boundary and the baseline set are the most useful parts, since a claimed saving can usually be traced to a narrow boundary or a weak baseline. Operators have a further interest. In a dense deployment the practical question is often not which tasks to offload but which edge sites to keep powered at a given hour, and the accounts of Step 3 make that trade visible.

Some simplification is unavoidable. Where a study cannot model the whole access network, the framework accepts a narrower boundary provided it is declared and the excluded terms are discussed qualitatively. Where task workloads are unknown, a parametric sweep is acceptable. What the framework does not accept is silence on these points.

5. Limitations and What Remains to Be Shown

The framework has not been exercised here on any scenario, so its usefulness is argued and not demonstrated. The obvious test is to re-express several published schemes within it, using a common scenario and a common energy boundary, and to see whether their relative merits change. A second test is whether the additional terms it insists on, particularly device waiting energy and idle infrastructure power, alter conclusions often enough to justify the extra modelling effort.

The component models carry their own limits. The quadratic energy-per-cycle law describes dynamic power in a single core and ignores leakage, heterogeneous cores and thermal throttling. The Shannon rate is an upper bound that real modulation and coding approach only within a margin. The base-station power model predates current hardware. Mobility is treated only through the delay account, although a user who changes cell during a task raises questions of service migration that the framework notes without resolving (Mach and Becvar, 2017). Privacy and security of offloaded data lie outside its scope altogether.

6. Conclusion

Energy-aware offloading in dense urban cells is a problem of shared resources, and its answer depends on whose energy is counted. The seven steps proposed here ask a study to declare its scenario and energy boundary, describe its tasks, write out every energy and delay term, formulate the problem openly, justify the solver by the timescale of the decision, compare against fixed baselines and probe its own assumptions. Adopted consistently, these steps would make results from different groups comparable. Whether they also change the conclusions that the field has reached is a question for the applications that should follow.

Conflict of interest

To be declared by the author(s) before publication.

Funding

To be declared by the author(s) before publication.

References

3GPP (2020). Study on channel model for frequencies from 0.5 to 100 GHz. Technical Report TR 38.901, Release 16. 3rd Generation Partnership Project, Sophia Antipolis.

Auer, G., Giannini, V., Desset, C., Godor, I., Skillermark, P., Olsson, M., Imran, M.A., Sabella, D., Gonzalez, M.J., Blume, O. and Fehske, A. (2011). How much energy is needed to run a wireless network? *IEEE Wireless Communications*, 18(5): 40-49. https://doi.org/10.1109/MWC.2011.6056691

Burd, T.D. and Brodersen, R.W. (1996). Processor design for portable systems. *Journal of VLSI Signal*

Processing, 13: 203-221.

Chen, X., Jiao, L., Li, W. and Fu, X. (2016). Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, 24(5): 2795-2808. https://doi.org/10.1109/TNET.2015.2487344

Cuervo, E., Balasubramanian, A., Cho, D., Wolman, A., Saroiu, S., Chandra, R. and Bahl, P. (2010). MAUI: Making smartphones last longer with code offload. *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 49-62. https://doi.org/10.1145/1814433.1814441

ETSI (2019). Multi-access Edge Computing (MEC); Framework and reference architecture. Group Specification ETSI GS MEC 003. European Telecommunications Standards Institute, Sophia Antipolis.

Huang, L., Bi, S. and Zhang, Y.-J.A. (2020). Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks. *IEEE Transactions on Mobile Computing*, 19(11): 2581-2593. https://doi.org/10.1109/TMC.2019.2928811

Kumar, K. and Lu, Y.-H. (2010). Cloud computing for mobile users: Can offloading computation save energy? *Computer*, 43(4): 51-56. https://doi.org/10.1109/MC.2010.98

Mach, P. and Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 19(3): 1628-1656. https://doi.org/10.1109/COMST.2017.2682318

Mao, Y., You, C., Zhang, J., Huang, K. and Letaief, K.B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4): 2322-2358. https://doi.org/10.1109/COMST.2017.2745201

Mao, Y., Zhang, J. and Letaief, K.B. (2016). Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE Journal on Selected Areas in Communications*, 34(12): 3590-3605. https://doi.org/10.1109/JSAC.2016.2611964

Neely, M.J. (2010). Stochastic Network Optimization with Application to Communication and Queueing Systems. Morgan & Claypool, San Rafael, CA.

Sardellitti, S., Scutari, G. and Barbarossa, S. (2015). Joint optimization of radio and computational resources for multicell mobile-edge computing. *IEEE Transactions on Signal and Information Processing over Networks*, 1(2): 89-103. https://doi.org/10.1109/TSIPN.2015.2448520

Satyanarayanan, M., Bahl, P., Cáceres, R. and Davies, N. (2009). The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Computing*, 8(4): 14-23. https://doi.org/10.1109/MPRV.2009.82

Sonmez, C., Ozgovde, A. and Ersoy, C. (2018). EdgeCloudSim: An environment for performance evaluation of edge computing systems. *Transactions on Emerging Telecommunications Technologies*, 29(11): e3493. https://doi.org/10.1002/ett.3493

Tran, T.X. and Pompili, D. (2019). Joint task offloading and resource allocation for multi-server mobile-edge computing networks. *IEEE Transactions on Vehicular Technology*, 68(1): 856-868. https://doi.org/10.1109/TVT.2018.2881191

You, C., Huang, K., Chae, H. and Kim, B.-H. (2017). Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Transactions on Wireless Communications*, 16(3): 1397-1411. https://doi.org/10.1109/TWC.2016.2633522

Zhang, K., Mao, Y., Leng, S., Zhao, Q., Li, L., Peng, X., Pan, L., Maharjan, S. and Zhang, Y. (2016). Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. *IEEE Access*, 4: 5896-5907. https://doi.org/10.1109/ACCESS.2016.2597169

Cite this article as: Jarrett N., Ortega J. and Desai M. (2026). Energy-Aware Task Offloading for Mobile Edge Computing in Dense Urban Cells: A Proposed Modelling and Evaluation Framework. *International Journal of Wireless Communication Networks and Mobile Computing*. 1(1), 15-20. doi: 10.62715/ijwcnmc.2026.v1.i1.pp15-20