

A DLT Framework for Service-Aware Link Adaptation in Quasi-Stationary Wireless Systems

Diego G. S. Pivoto, Felipe A. P. de Figueiredo, and Luciano L. Mendes

Abstract—Distributed ledger technology (DLT) can enable programmable and auditable markets for radio capabilities. This paper presents an end-to-end DLT framework that connects smart-contract-based service acquisition to policy activation on a real software-defined radio (SDR) platform operating outside a Third Generation Partnership Project (3GPP) stack. A base station (BS) acts as service provider and a user equipment (UE) as consumer; an Ethereum-compatible ledger, a Representational State Transfer (REST) application programming interface (API), receipt/event validation, and synchronized radio control form the transaction-to-radio bridge. As the instantiated radio service, the SDR platform implements a bit error rate (BER)-constrained service-aware link adaptation (SALA) mechanism based on contextual bandits. For the experimental validation considered in this work, the link is characterized beforehand over 41 signal-to-noise ratio (SNR) contexts, generating three service-specific SNR-to-modulation and coding scheme (MCS) look-up tables (LUTs) associated with distinct BER requirements and stored locally at both endpoints. A radio-over-fiber (RoF)-assisted experiment shows that purchasing the stringent policy activates the intended LUT at the four contexts where it differs from the default policy, increasing stringent-service coverage from 37/41 contexts (90.2%) to 41/41 (100%), a 9.8-percentage-point gain, with localized goodput reductions. The same service-identifier abstraction can support broader DLT markets for radio-control and resource-allocation policies.

Keywords—DLT, smart contracts, service-aware link adaptation, SNR-MCS mapping, software-defined radio, radio-over-fiber, quasi-stationary links.

I. INTRODUCTION

Future wireless networks are expected to support flexible service activation according to application requirements through software-defined networking (SDN), service-oriented architecture (SOA), and artificial intelligence (AI) techniques [1], [2]. Such operation requires trusted mechanisms for service registration, authorization, payment validation, and automatic radio reconfiguration.

Distributed ledger technology (DLT), commonly implemented through blockchain and smart contracts, provides programmable rules and auditable transaction records for service-market models [3], [4]. A smart contract can validate a service acquisition and authorize a radio access network (RAN) controller to activate the corresponding functionality.

Adaptive modulation and coding (AMC) is one functionality that can benefit from this model [13]. Conventional link adaptation selects a modulation and coding scheme (MCS) mainly from a signal-to-noise ratio (SNR)-to-MCS mapping. A single mapping, however, does not inherently distinguish traffic classes with different residual bit error rate (BER)

requirements. Sensor and Internet of Things (IoT) traffic can be more error-tolerant than multimedia or critical traffic, while applying the most conservative policy to every flow can unnecessarily reduce goodput.

Prior DLT networking work has mainly addressed service and slice orchestration. The 5G-as-a-Service (5GaaS) architecture integrates DLT and smart contracts with a decentralized network-slice marketplace, billing, and lifecycle management [5], while Bukhari *et al.* combine smart contracts and meta-learning for fine-grained service-level agreement and resource management in a 6G slice [6]. At the radio level, conventional link adaptation optimizes channel-dependent MCS decisions [13]. The SDR-based experimental platform and its previously implemented goodput-driven contextual-bandit (CB) AMC mechanism are described in [12]. Building on this environment, the radio-control functionality considered in this work is a BER-constrained service-aware link-adaptation mechanism based on contextual bandits and developed for the considered SDR platform. It characterizes the link beforehand and generates service-specific SNR-MCS LUTs according to the BER requirement of each service. However, the prior approaches do not provide a purchase-and-authorization workflow that connects an on-chain service acquisition to the synchronized activation of a service-specific MCS policy on a real radio platform. The novelty here is an end-to-end provider-consumer transaction-to-radio bridge.

For the experimental validation considered in this work, the SDR platform implements a BER-constrained service-aware link-adaptation mechanism based on contextual bandits. The link is characterized beforehand over its operating SNR range, resulting in three service-specific SNR-MCS look-up tables (LUTs) that are stored locally at both endpoints. Marketplace operation begins with these available policies: the user equipment (UE) purchases one service, the base station (BS) validates the on-chain event, and both endpoints activate the corresponding LUT, while the ledger stores only the service identifier, payment, and authorization record. The contributions are: (i) a complete DLT service market with a service catalog, smart-contract payment, auditable purchase records, and explicit provider/consumer roles suitable for decentralized multi-entity deployment; (ii) a blockchain-to-radio control bridge that integrates validated on-chain events with synchronized activation, at the BS and UE, of a BER-constrained service-aware link-adaptation policy based on contextual bandits and implemented on a real SDR platform outside a Third Generation Partnership Project (3GPP) stack; (iii) experimental validation of DLT-triggered switching from the default Service 1/2 LUT to the stringent Service 3 LUT at the four SNR contexts in which the stricter BER requirement changes the MCS decision, increasing stringent-service

The authors are with the National Institute of Telecommunications (Inatel), Santa Rita do Sapucaí, MG 37536-001, Brazil (diegopivoto@gea.inatel.br, felipe.figueiredo@inatel.br, luciano1@inatel.br).

coverage from 90.2% to 100%; and (iv) a service-identifier abstraction that can extend the market beyond link adaptation to other radio-control and resource-allocation policies.

II. SYSTEM MODEL

A. DLT-Enabled Service Orchestration

The BS offers a set of purchasable SALA policies. Each service $s \in \mathcal{S}$ is associated with a target BER β_s , a price, and a local LUT $\mathcal{L}_s$. A private Ethereum-compatible network represents the BS and UE as blockchain accounts, while a smart contract stores service identifiers, prices, and purchase records [10]. For each acquisition, the contract checks the transferred value, registers the buyer, seller, selected service, UE identifier, and timestamp, and emits an event that can be independently validated.

A Representational State Transfer (REST) application programming interface (API) exposed by the BS agent connects the UE DLT client to the radio-control software. The UE retrieves the service list through this interface, signs and submits the smart-contract call, and sends the resulting transaction hash to the BS. The BS retrieves the receipt from the ledger, decodes the emitted event, and verifies transaction status, service, UE identifier, seller, and paid value. A valid event authorizes the BS and UE agents to select and activate the same previously generated, locally stored LUT. Thus, the ledger provides acquisition and authorization, whereas policy storage and execution remain in the SDR platform.

Although the evaluated service object is a link-adaptation LUT, the contract/API boundary is resource-agnostic. A validated service identifier can map to other locally enforced functions, such as scheduler shares, time–frequency allocations, spectrum-access profiles, or transmit-power policies. This allows radio nodes or infrastructure owners to act as providers and consumers without placing the implementation or radio measurements on-chain.

B. Precomputed BER-Constrained SALA Policies

At runtime, the DLT framework authorizes and synchronizes the activation of the LUT associated with the purchased service. The SDR testbed and its previously implemented goodput-driven contextual-bandit AMC mechanism are described in [12]. The radio-control functionality considered in this work is a BER-constrained service-aware link-adaptation mechanism based on contextual bandits and developed for this SDR platform. Before DLT operation, this mechanism is executed across 41 SNR contexts from 13.5 to 33.5 dB in 0.5-dB steps and the 17 supported MCS actions. For context x , action a , and service s , measured goodput and post-decoding BER define:

$$R_s(x, a) = \begin{cases} \bar{G}(x, a)/G_{\max}, & \hat{p}(x, a) \leq \beta_s, \\ -1, & \hat{p}(x, a) > \beta_s, \end{cases} \quad (1)$$

where $\bar{G}(x, a)$ is the mean goodput, $G_{\max}$ is the normalization factor, and $\hat{p}(x, a)$ is the measured BER. After the prior characterization, the frozen policy is:

$$\mathcal{L}_s(x) = \min \left(\arg \max_{a \in \mathcal{A}} R_s(x, a) \right), \quad (2)$$

TABLE I: SALA policies offered by the DLT service.

Service	Representative Class	DSCP	BER Threshold
1	Relaxed (sensor/IoT)	Best Effort (BE)	10^{-3}
2	Intermediate (multimedia)	Assured Forwarding 21 (AF21)	10^{-5}
3	Stringent (critical)	Expedited Forwarding (EF)	1.3×10^{-8}

where $R_s(x, a)$ is the estimated service-dependent reward and exact ties favor the lower MCS. The three resulting LUTs are stored locally at the BS and UE before the DLT marketplace becomes operational.

For the implemented link, Services 1 and 2 produce the same LUT entries throughout the 41-context range because their goodput-maximizing actions satisfy both less restrictive BER limits. Service 3 differs only at 16.5, 20.0, 20.5, and 22.5 dB, i.e., four transition contexts where the MCS used by Services 1/2 still satisfies 10^{-3} and 10^{-5} but exceeds the 1.3×10^{-8} limit. In those contexts, the Service 3 LUT selects the immediately lower MCS. These three previously generated policies constitute the radio services exposed through the DLT market.

The differentiated services code point (DSCP) labels the representative class. Table I summarizes the policies exposed by the smart contract. They are relative link-level reliability tiers rather than universal end-to-end application guarantees.

C. Quasi-Stationary Operating Scope

The target class comprises fixed point-to-point radio links, including fixed wireless access, rural connectivity, industrial or campus links, and remote-radio deployments in which end-point geometry, antenna alignment, hardware configuration, and the dominant propagation regime vary slowly. Under this bounded quasi-stationary envelope, the characterization cost can be amortized over an extended period, and runtime execution becomes a deterministic LUT lookup. Persistent departure from the characterized SNR range requires reassessment or recalibration.

III. EXPERIMENTAL SETUP

A. SDR and RoF-Assisted Topology

The implementation uses the BS and UE labels of the SDR platform to denote, respectively, the downlink transmitter/controller and the receiver/feedback endpoint. The evaluated connection is therefore described as a point-to-point SDR link, in which the underlying platform implements an ultra-high frequency (UHF) TV white space (TVWS) radio and supports direct over-the-air radio frequency (RF) operation [9], [12]. For controlled laboratory experiments, the DLT evaluation reported herein adopts a radio-over-fiber (RoF)-assisted realization, providing tighter control over the operating conditions and transmission-gain variation across the wider SNR range supported by the link-adaptation controller, while also supporting a 20 km laboratory link span between the endpoints.

In this realization, each SDR RF port is connected to an RoF module. The electrical RF waveform is converted to the optical domain, transported through 20 km of single-mode

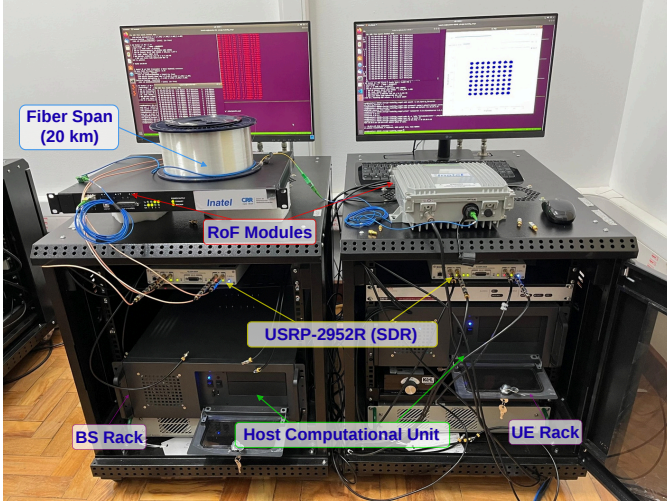


Fig. 1: SDR-based BS-UE experimental topology using RoF modules and a 20-km optical-fiber span. The virtual IP tunnel is established over the RF/RoF link.

TABLE II: Main experimental parameters.

Parameter	Configuration
Radio roles	BS transmitter/controller; UE receiver/feedback
SDR/synchronization	NI-USRP-2952R; global positioning system (GPS)
Waveform/duplexing	Generalized frequency-division multiplexing (GFDM); frequency-division duplexing (FDD)
Bandwidth/frequencies	24 MHz; 560 MHz uplink, 680 MHz downlink
MCS set	17 levels, quadrature phase-shift keying (QPSK) to 256-ary quadrature amplitude modulation (256-QAM)
RoF path	Analog RF waveform over 20 km single-mode fiber
Prior SALA stage	41 contexts, 13.5–33.5 dB; three frozen local LUTs
DLT activation contexts	16.5, 20.0, 20.5, and 22.5 dB
Per-state radio test	15 s goodput followed by 45 s BER
DLT/control software	Ganache, Solidity, Remix, Web3.py, REST API
Service price	3 Ether per SALA policy

fiber, converted back to RF, and delivered to the opposite SDR. The fiber carries the analog radio waveform while preserving the complete physical-layer (PHY) and medium access control (MAC) processing chain. Once the RF/RoF link is established, the SDR platform creates a virtual Internet Protocol (IP) tunnel between the BS and UE hosts over this communication link. Inter-host REST messages and the UE-originated blockchain transaction are carried as IP traffic through this tunnel, whereas receipt and event queries between the BS agent and Ganache remain local to the BS host.

Figure 1 summarizes the experimental topology, including the BS-UE SDR testbed, the RoF modules and 20-km optical-fiber span, and the virtual IP tunnel used for network-layer data and control exchanges between the hosts.

Table II summarizes the parameters directly relevant to the DLT-to-radio validation. The four DLT activation contexts are exactly those at which the Service 3 LUT differs from the default Services 1/2 mapping.

B. DLT and Radio-Control Agents

Ganache instantiates the private ledger and initializes the BS and UE test accounts with 100 Ether (ETH) each [7]. Before runtime, the Solidity smart contract is compiled and

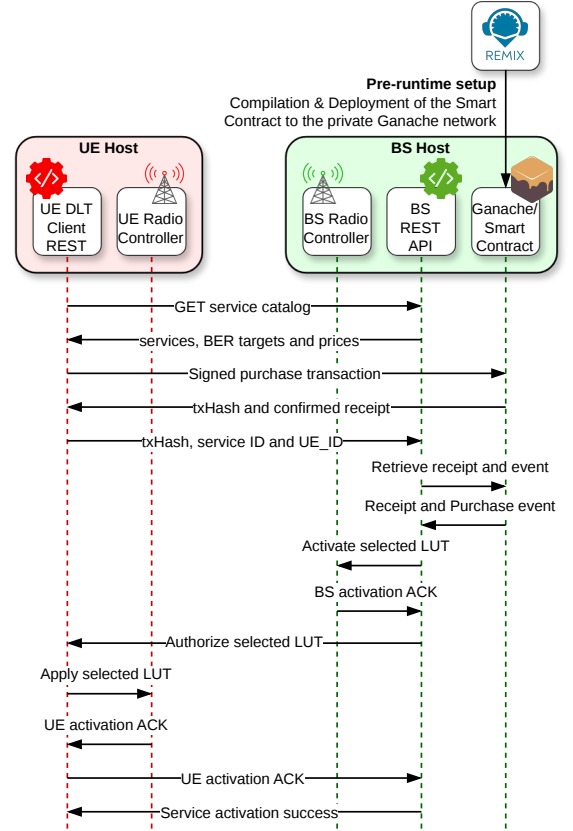


Fig. 2: Runtime message exchange for DLT-based service acquisition and activation of a precomputed SALA policy. The UE submits the signed purchase to Ganache, the BS validates the receipt and emitted event through Web3.py, and both radio endpoints activate the same locally stored LUT.

deployed to the Ganache network through the Remix integrated development environment (IDE) [8]. Two Python agents implement the transaction-to-radio bridge. The UE DLT client retrieves the catalog from the BS REST API, selects a service, queries its price, signs and submits the contract call, waits for block confirmation, and sends the transaction hash, service identifier, and UE identifier to the BS. The BS agent retrieves and validates the receipt and emitted event through Web3.py, maps the service identifier to the corresponding local LUT, applies it at the BS, and authorizes the UE to apply the same LUT. These LUTs are stored locally before runtime; neither their contents nor radio measurements are placed on-chain.

Figure 2 summarizes this runtime message exchange. After the UE reports completion of its local radio update, the BS REST API returns the final service-activation result to the UE DLT client.

Algorithm 1 formalizes service acquisition, on-chain validation, and synchronized activation of the LUTs already available at both radio endpoints.

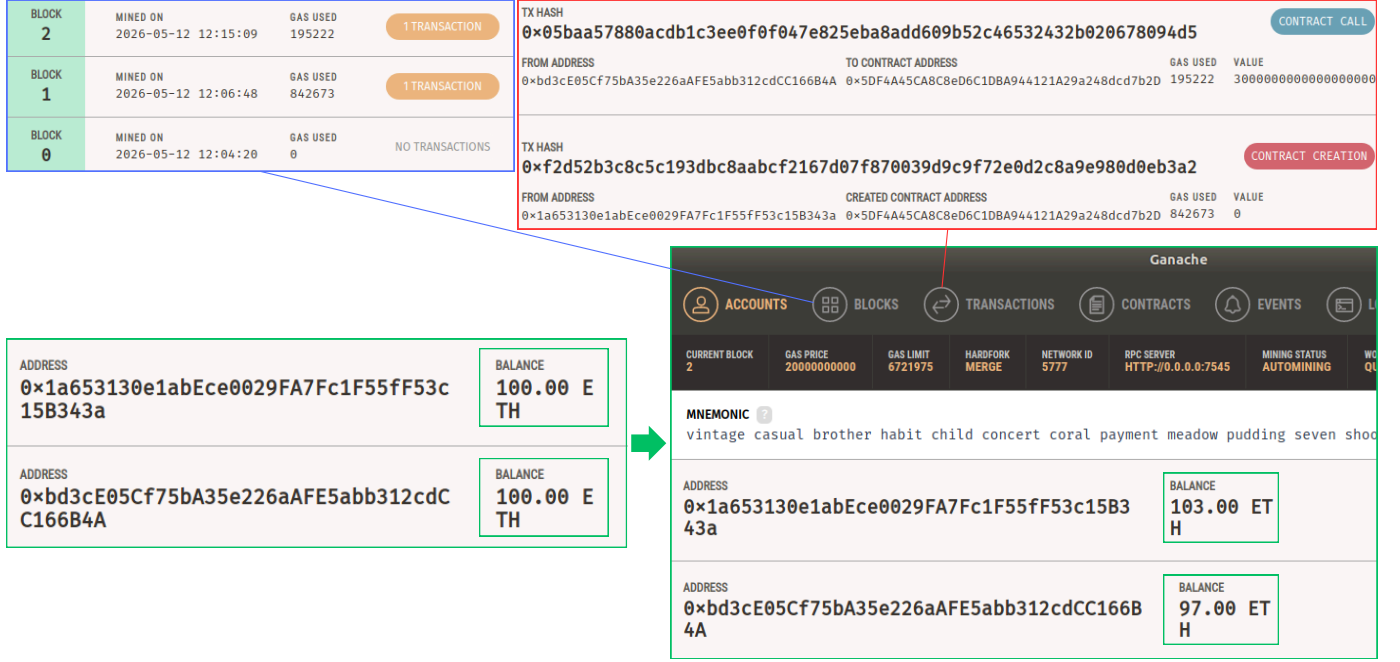


Fig. 3: Private Ganache ledger used for contract deployment and Service 3 acquisition. The accounts, blocks, and transaction record provide the auditable evidence subsequently checked by the BS agent.

Algorithm 1 DLT acquisition and activation of a precomputed SALA policy

Require: $\mathcal{S} = \{1, 2, 3\}$, $\beta = \{10^{-3}, 10^{-5}, 1.3 \times 10^{-8}\}$
Require: price $P_s = 3$ ETH and frozen local LUTs $\{\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3\}$

- 1: UE requests the service catalog from the BS REST API
- 2: BS returns $\mathcal{S}$, β , P_s , and descriptions
- 3: UE selects s^* , queries P_{s^*} , and signs $(s^*, \text{UE_ID})$
- 4: UE submits the contract call with value P_{s^*}
- 5: Contract validates payment, records the purchase, and emits event E
- 6: UE obtains txHash after block confirmation and sends $(\text{txHash}, s^*, \text{UE_ID})$ to the BS
- 7: BS retrieves the receipt and decodes E
- 8: **if** status, buyer, seller, service, UE identifier, and value are valid **then**
- 9: BS maps and applies local $\mathcal{L}_{s^*}$ and restarts/updates its radio stack
- 10: BS authorizes the UE to activate the same local $\mathcal{L}_{s^*}$
- 11: UE applies $\mathcal{L}_{s^*}$, restarts/updates its radio stack, and reports completion
- 12: BS returns service s^* activated to the UE
- 13: **else**
- 14: Reject activation and preserve the current LUT
- 15: **return** transaction or validation failure
- 16: **end if**

IV. RESULTS AND DISCUSSION

A. DLT Transaction Validation

Following the runtime message sequence shown in Figure 2, Figure 3 records the two relevant ledger operations. Block 1 deploys the contract from the BS account, and Block 2 records the UE call that purchases Service 3 for 3 ETH. The displayed balances change from 100/100 ETH to 103/97 ETH for the BS/UE accounts, respectively. The UE then sends the transaction hash, service, and UE identifier to the BS agent. Receipt and event validation confirm the buyer, seller, service, UE identifier, transferred value, and successful status before any radio change is authorized.

Together, the smart contract, event validation, the Web3.py-

enabled BS agent, the REST interface, and radio-control calls implement the complete transaction-to-radio framework. The ledger establishes an auditable provider-consumer record, while the selected policy and radio measurements remain local to the SDR nodes. The Ganache record represents one Service 3 purchase; after activation, that policy remains selected while the operating point is stepped through the four test contexts. With a fixed 3-ETH price, n successful purchases would produce displayed balances $100 + 3n$ ETH and $100 - 3n$ ETH at the BS and UE, respectively, excluding gas and subject to sufficient UE funds. The same validated identifier-to-control mapping can support future markets for radio-resource allocation and other locally enforced functions; long-duration consensus and multi-entity scalability remain outside this local-ledger experiment.

B. DLT-Triggered Reconfiguration at the Four Critical Contexts

Before the purchase, the Service 1 LUT is the platform default. Because the Service 1 and Service 2 LUTs coincide over the characterized range, this initial state also represents the MCS selected for Service 2. The link is first operated at 16.5, 20.0, 20.5, and 22.5 dB with this default LUT. The UE then performs the single Service 3 purchase shown in Fig. 3; after Algorithm 1 validates the transaction and activates $\mathcal{L}_3$, the same four contexts are revisited.

Each selected context-MCS state is held for 60 s: mean SNR and downlink goodput are measured with *iperf3* during the first 15 s, followed by 45 s of decoded reference-bit and error counting for BER. Table III reports the before/after results. For the post-purchase states, no errors were observed; hence, the BER entries are the corresponding one-sided 95% upper bounds rather than zero-point estimates.

TABLE III: Radio behavior before and after the validated Service 3 purchase at all four contexts where its precomputed LUT differs from Services 1/2. Each state uses 15 s of goodput measurement and 45 s of BER measurement.

SNR (dB)	Before purchase: default Service 1/2 policy			After validation: Service 3 policy		
	MCS	Goodput (Mbps)	BER	MCS	Goodput (Mbps)	BER / 95% upper bound
16.5	8	50.84	2.49×10^{-8}	7	43.38	$< 1.53 \times 10^{-9}$
20.0	10	71.24	5.09×10^{-8}	9	62.47	$< 1.07 \times 10^{-9}$
20.5	10	69.74	1.43×10^{-8}	9	63.52	$< 1.05 \times 10^{-9}$
22.5	12	86.44	1.52×10^{-8}	11	76.30	$< 8.73 \times 10^{-10}$

The first block of Table III represents both Services 1 and 2, while the second represents Service 3. Before the purchase, every selected MCS satisfies the Service 1 and Service 2 thresholds but violates the stringent 1.3×10^{-8} requirement. After the validated purchase, the active LUT reduces the MCS by one level at each context, and every resulting BER bound lies below the Service 3 limit. The corresponding goodput reductions are 7.46, 8.77, 6.22, and 10.14 Mbps and are confined to these four decision-critical contexts.

Across the complete 41-context operating range, the default Services 1/2 mapping satisfies the Service 3 requirement at 37 contexts (90.2%). DLT-triggered activation of the purchased Service 3 LUT resolves the four uncovered contexts and reaches 41/41-context coverage (100%), a gain of 9.8 percentage points. The transaction therefore has a direct radio-level consequence: it authorizes and activates the LUT previously generated by the BER-constrained contextual-bandit SALA mechanism, trading goodput only where required to make the stringent service feasible while leaving the other 37 MCS decisions unchanged.

Therefore, the implementation logs transaction completion, validation, LUT selection, and radio-stack restart, and future instrumentation should report mining, receipt validation, API exchange, and stack-restart components separately.

V. CONCLUSION

This paper presented an end-to-end DLT marketplace and transaction-to-radio control framework for a real SDR platform outside the 3GPP stack. The proposed framework connects a decentralized service-acquisition mechanism to a real SDR radio-control functionality, in which a BER-constrained contextual-bandit SALA mechanism generates service-specific LUTs beforehand, and the DLT layer determines which policy is authorized and activated on demand. The smart contract, provider/consumer accounts, auditable payment and event records, the Web3.py-enabled BS agent, the BS REST API, and synchronized BS/UE control jointly connect a validated purchase to activation of a locally available radio policy. In the SALA use case, purchasing Service 3 changed the MCS at the four decision-critical contexts and increased stringent-service coverage from 37/41 (90.2%) to 41/41 contexts (100%), a 9.8-percentage-point gain. Although experimentally instantiated with link-adaptation LUTs, the service-identifier abstraction can also support markets for scheduling, spectrum, power, and other radio-resource policies. Future work will migrate the controlled Ganache setup to a distributed permissioned

DLT network, and evaluate multi-provider operation, different RF scenarios, and departures from the characterized quasi-stationary envelope.

ACKNOWLEDGMENTS

This work was partially funded by CNPq (302085/2025-4, 306199/2025-4), by FAPEMIG (PPE-00124-23, APQ-04523-23, APQ-05305-23, RED-00194-23, APQ-03162-24), by the Brasil 6G project (01245.020548/2021-07), by project XGM-AFCCT-2024-9-1-1 supported by xGMobile-EMBRAPII-Inatel with financial resources from the PPI IoT/Manufatura 4.0 from MCTI (052/2023), and by FINEP (No. 1060/2, contract 01.25.0883.00).

REFERENCES

- [1] M. Beshley, N. Kryvinska, et al., "Traffic engineering and QoS/QoE supporting techniques for emerging service-oriented software-defined network," *Journal of Communications and Networks*, vol. 26, no. 1, pp. 99–114, 2024, doi: 10.23919/JCN.2023.000065.
- [2] D. G. S. Pivoto, F. A. P. de Figueiredo, et al., "A comprehensive survey of machine learning applied to resource allocation in wireless communications," *IEEE Communications Surveys & Tutorials*, vol. 28, pp. 1986–2053, 2026, doi: 10.1109/COMST.2025.3552370.
- [3] X. Wang, X. Ren, C. Qiu, Z. Xiong, H. Yao, and V. C. M. Leung, "Integrating edge intelligence and blockchain: What, why, and how," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2193–2229, 2022, doi: 10.1109/COMST.2022.3189962.
- [4] A. Abubhashim and C. C. Tan, "Smart contract designs on blockchain applications," in *Proc. 2020 IEEE Symposium on Computers and Communications (ISCC)*, 2020, pp. 1–4, doi: 10.1109/ISCC50000.2020.9219622.
- [5] K. Rasol, A. Egio, M. Catalan-Cid, L. Lossi, H. Simeão, and S. Siddiqui, "5GaaS: DLT and smart contract-based network slice management in a decentralized marketplace," in *Proc. 20th International Conference on Network and Service Management (CNSM)*, 2024, pp. 1–7, doi: 10.23919/CNSM62983.2024.10814406.
- [6] S. Bukhari, K. Sharif, L. Zhu, C. Xu, F. Li, and S. Biswas, "Dynamic fine-grained SLA management for 6G eMBB-plus slice using mDNN and smart contracts," *IEEE Transactions on Services Computing*, vol. 17, no. 6, pp. 3499–3512, 2024, doi: 10.1109/TSC.2024.3453709.
- [7] Truffle Suite, "Ganache: One click blockchain," 2026. [Online]. Available: <https://archive.trufflesuite.com/ganache/>. Accessed: May 13, 2026.
- [8] Remix Project, "Remix IDE: Smart contract development," 2026. [Online]. Available: <https://remix.ethereum.org/>. Accessed: May 13, 2026.
- [9] M. S. B. Cunha, J. de C. Matos, et al., "Coexistence analysis of DTV signals with 4G, 5G and 5G-RANGE signals in TVWS," *IEEE Transactions on Vehicular Technology*, vol. 75, no. 4, pp. 6705–6716, 2026, doi: 10.1109/TVT.2025.3622504.
- [10] P. Hegedus, "Towards analyzing the complexity landscape of Solidity-based Ethereum smart contracts," in *Proc. 2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, 2018, pp. 35–39.
- [11] A. Ghosh, A. Sankaraman, and K. Ramchandran, "Model selection for generic contextual bandits," *IEEE Transactions on Information Theory*, vol. 70, no. 1, pp. 656–675, 2024, doi: 10.1109/TIT.2023.3298103.
- [12] D. G. S. Pivoto, J. S. Ferreira, et al., "Contextual bandit-based AMC for quasi-stationary wireless backhaul links on SDR platforms," *IEEE Open Journal of the Communications Society*, pp. 1–1, 2026, doi: 10.1109/OJCOMS.2026.3692574.
- [13] C. Zhang, E. Medeiros, et al., "Robust fronthaul-aware link adaptation for UAV communication with lossy wireless fronthaul," *IEEE Communications Letters*, vol. 29, no. 10, pp. 2263–2267, 2025, doi: 10.1109/LCOMM.2025.3592520.