Distributed Trust and Reputation Management for Future Wireless Systems

Dev P. Singh, *Student Member, IEEE*, Kevin W. Sowerby, *Senior Member, IEEE*, and Andrew C. M. Austin, *Member, IEEE*

Abstract—**Trust lies at the center of the paradigm shift required to realize ultra-dense networks (UDN) needed by future radio communication systems. We propose a distributed, three-layer, trust-based hardware sharing scheme between operators that overcomes the limitations of a single operator owned monolithic network. Our system scales to tens of thousands of operators without requiring explicit contracts between operators, or between operators and user-devices. User-devices in turn are free to requisition the services of any available hardware. This is achieved by abstracting the communication process as a transaction, and casting it within a Distributed Ledger Technology (DLT) framework paired with an efficient, fault-tolerant, distributed consensus protocol. A trust model associates a behavioral measure with each hardware device which signals its reliability, as well as its payoff. The proposed system offers multiple advantages for users, operators, and regulators.**

I. Introduction

Radio communication systems of the future will move away from fixed infrastructure providers and static contracts, and towards multi-tenanted systems featuring actively negotiated terms of service. Such a setting will feature thousands of hardware providers, including traditional and non-traditional cellular operators. Network hardware and user-equipment (UE), will freely interact with each other to set-up, execute and resolve communication tasks, without any static prior contracts. Consequently, the reliability of UE, network hardware and service provisioning will need to be actively considered in the setup and execution of communication tasks. In a traditional cellular system, trust management occurs through periodically revised, static contracts. An assumption of complete trust holds during a contractual period. Such an approach is not suitable for ultra-dense massively shared network hardware (SNH) systems due to the significant overheads introduced by explicit contracts between all UE and SNH, as well as between SNH belonging to different operators. Additionally, the reluctance of rival operators to freely share information, or make their proprietary systems public may invalidate the perfect trust assumption. Finally, repeat interactions between NH and UE in an ultra-dense system may not occur frequently enough to generate statistically significant reliability measures.

Therefore UE and SNH need to resolve terms of interaction within their local context, based on uncertain or incomplete information of each other. This is precisely the type of conditions under which trust enables distributed decision making within social systems. Trust within such settings is restricted to a given activity, applied to direct interactions, and based on the discretion of the trusting party. Consequently, social-trust based measures need to correlate the trust rating of a device with it's actual performance at fulfilling it's stated role. For instance, a UE in a shared setting is expected to pay for the services it consumes, whereas a SNH device is expected to adhere to the terms of the service level agreement (SLA). Experiences and outcomes perform a central role in trust based decision-making. Therefore a trust based system needs to record the history of relevant interactions. Furthermore, in order for these records to form a meaningful basis, they must meet information security requirements such as consistency, availability, and immutability.

A massive SNH system that replaces fixed subscriptions between users and operators with dynamic, locally initiated, trust-based contracts, mandates a distributed notion of trust.

Authorized licensed use limited to: University of Auckland. Downloaded on June 28,2022 at 03:48:26 UTC from IEEE Xplore. Restrictions apply.

This is to ensure the computation, update, and propagation of trust occurs without centralized co-ordination, or pre-trusted relationships/ trusted third parties.

We propose a trust based SNH system that meets the requirements of distributed setup, execution, and control of network activity. Our system consists of a Distributed Ledger Technology (DLT) [1] component, paired with a Byzantine Fault Tolerant (BFT) [2] consensus protocol, and a distributed trust model. The trust model uses a combination of behavioral and commodified trust to translate native DLT trust structures into communication system equivalents.

Our system offers several advantages for SNH operators and UE, including infrastructure independence, dynamic contracts, and a practical, low-cost pathway to network densification. Additionally, it provides regulators with a distributed framework to implement policies related to radio-resource management, fair-use, and energy-efficiency.

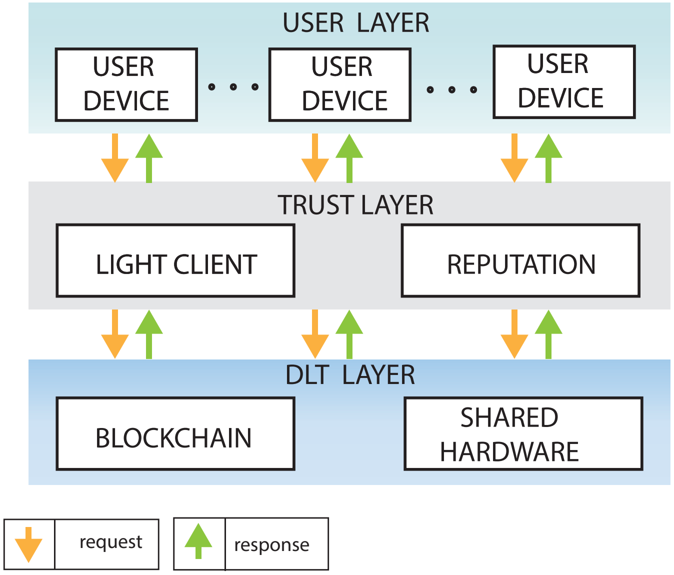


Fig. 1: Three-layer model of the proposed system

II. Related Work

Existing work adopts one of two main approaches to addressing the high-frequency, ultra-dense network deployment problem. The first of these is framed as the Network Embedding Problem (NEP) [3]. NEP invokes software defined control, and service oriented architecture to abstracts hardware resources into groups of network functions. Each such logical partitioning, termed a network slice [3], is then allocated to service incoming user-requests.

The second widely used approach focuses on the performance of infrastructure sharing in UDNs. Stochastic Geometry is used to model the distribution of user equipment and operator hardware. Performance metrics such as Signal to Interference Noise Ratio (SINR), and Outage Probability derived from these models [4] guide deployment.

Both of these approaches, assume single-tenant systems and focus on sharing of resource-blocks rather than hardware. Consequently, most works exploring resource-sharing are formulated in a centralized setting. Our work employs hardware-sharing, extended to a multi-tenant setting, featuring thousands of operators committing network hardware, and without fixed-contracts between UE and SNH. Additionally, the allocation, pricing and management of shared network functions is implemented in a fully distributed manner. Recent works have looked at DLT based resource sharing in Mobile Edge Computing [5], and Industrial Internet-of-Things [6] settings, focusing on obfuscation of identity and routing topology respectively.

Distributed systems have been extensively researched. This has produced a rich family of system specifications, and protocols. However, until the advent of DLT most such systems were relegated to tightly constrained academic or private settings. DLT represents the only production-grade, large-scale, distributed, fault-tolerant system. The central idea originates from Multi-Party Computation (MPC) [7] which uses State Machine Replication (SMR) to synchronize the states of multiple servers concurrently responding to service requests from clients. The service is abstracted as a state machine replicated across the servers. Paxos [8] style protocols were among the first to solve the SMR distributed consensus problem, by invoking the atomic broadcast [2] primitive.

An SMR service is termed fault-tolerant if it progresses despite crashes or corruptions

Authorized licensed use limited to: University of Auckland. Downloaded on June 28,2022 at 03:48:26 UTC from IEEE Xplore. Restrictions apply.

among a minority quorum of replicas. Such protocols proceed in rounds of message passing such that at the end of a round, all honest parties are guaranteed to commit to the same set of values. In the case of DLT systems, the value being committed to is a batch of transactions, and all correct participants commit by appending the batch to their local ledger as a new DLT block. Each round consists of protocol participants broadcasting their proposals, voting on each other's proposals, and using majority vote to decide. The voting mechanism is generally implemented using threshold cryptosystems [9] [10].

DLT systems generate leaderless agreement among nodes on the state of the DL without needing a trusted third party, such as a bank or central server [11]. Such systems are thus well suited to a massively shared, decentralized communication system featuring thousands of traditional and non-traditional operators. Bitcoin, the pioneering DLT system introduced Proof-of-Work (PoW) [11] style protocol to solve consensus. PoW protocols have unacceptably high energy-consumption due to the computational cost associated with securing the public or permissionless DL against malicious actors.

However, in a permissioned DLT network participants are not anonymous, allowing PoW can be substituted by more efficient, provably-secure protocols from the field of fault-tolerant distributing computing [12] [13].

Distributed consensus protocols adopt different notions of reliability, and are classified accordingly. A protocol is termed synchronous if message delivery occurs within bounded time. Whereas asynchronous variants feature no external timing reference, and employ randomisation protocols such as the coin-tossing [2] to measure progress. Messages in a radio communication system may be lost, delayed or arrive out of order due to noise, interference, channel state variability, and other such sources of nondeterminism. Therefore, asynchronous agreement protocols are better suited to specify such systems.

III. System Model

Our proposed system is abstracted as a three-layer model, as shown in Figure 1, with the lower layers responding to service requests from upper layers. The lowest DLT layer comprises the blockchain framework. A UE in the system interacts with the shared hardware through the DLT light client, and the reputation model. The DLT light client manages a UE's past transactions, available balances, transaction composition, and third-party interactions. The trust model is employed as a decision making device by both SNH and UEs.

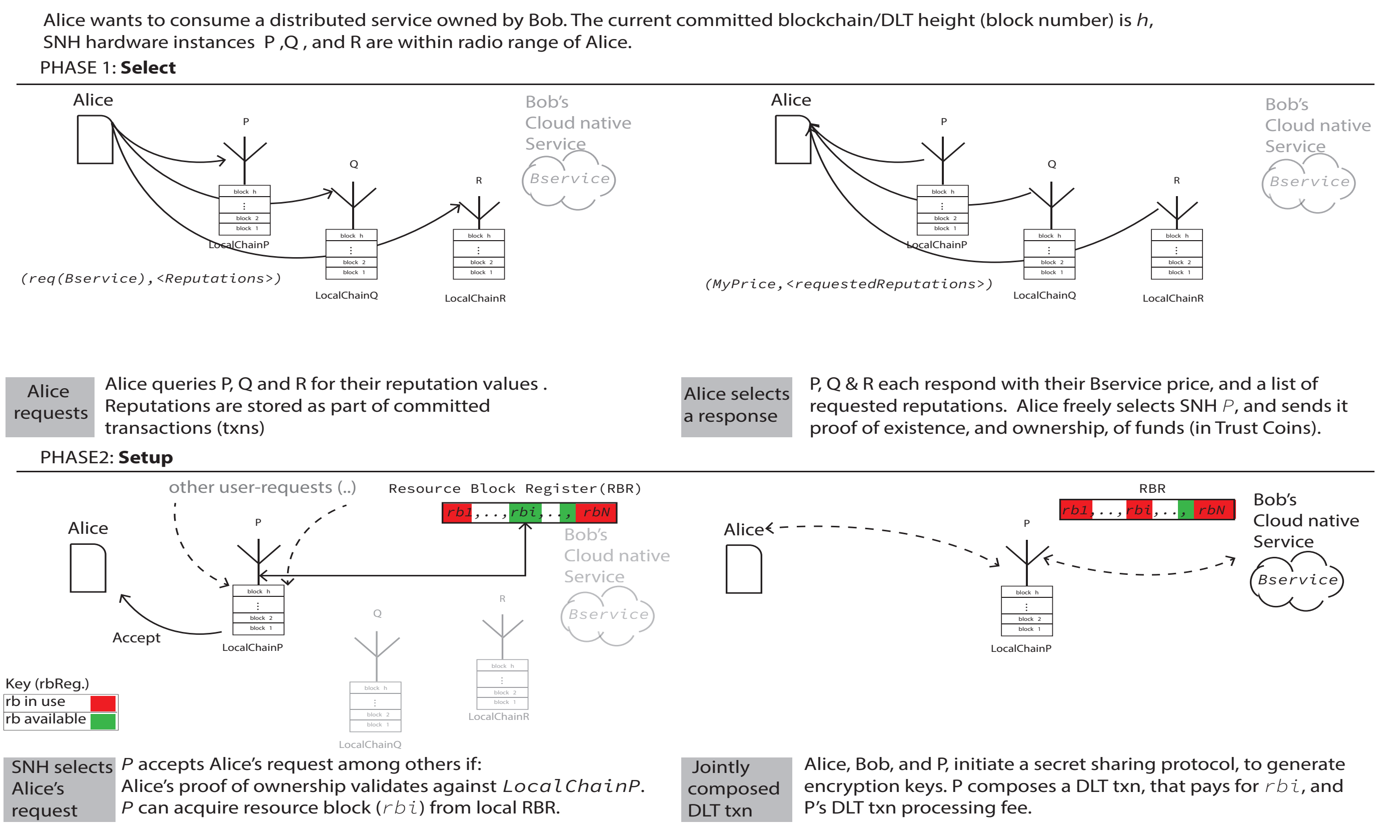
Figures 2 and 3 illustrate the different stages of communication in our system, as an interaction between a UE and SNH. A UE initiates the process whenever it needs to execute a communication task over the network. Compared to traditional dedicated hardware, UE-SNH interactions involve several additional stages concerned with setup, execution, and control.

Trust management in our system involves representing, computing, updating and storing trust. Trust is represented as a behavioral measure correlated to a SNH device's past record of fulfilling its stated function. This takes the form of a reputation rating assigned to each device. Every interaction between a SNH and UE, encoded as DLT transaction, triggers a change in SNH reputation. Updated reputations are included within the DLT transaction, and are verified via the consensus protocol that adds new transactions to the DLT ledger. The distributed nature of the ledger, ensures trust values reliably propagate through the system, and are readily available. An SNH device in our system, acting as a DLT consensus node, is capable of self-managing all aspects of their operation. A software based formulation of this node ensures SNH devices can easily be switched between shared and private modes, at little cost to the operator. This flexibility reduces the barrier to entry for a SNH device wanting to join the system, thereby enhancing scalability.

A. Trust Model

Trust modeling has traditionally occupied the realm of soft-security, as an alternative to hard-security measures based on cryptography.

Authorized licensed use limited to: University of Auckland. Downloaded on June 28,2022 at 03:48:26 UTC from IEEE Xplore. Restrictions apply.



Alice wants to consume a distributed service owned by Bob. The current committed blockchain/DLT height (block number) is *h*, SNH hardware instances P ,Q , and R are within radio range of Alice.

PHASE 1: **Select**

Alice requests: Alice queries P, Q and R for their reputation values . Reputations are stored as part of committed transactions (txns)

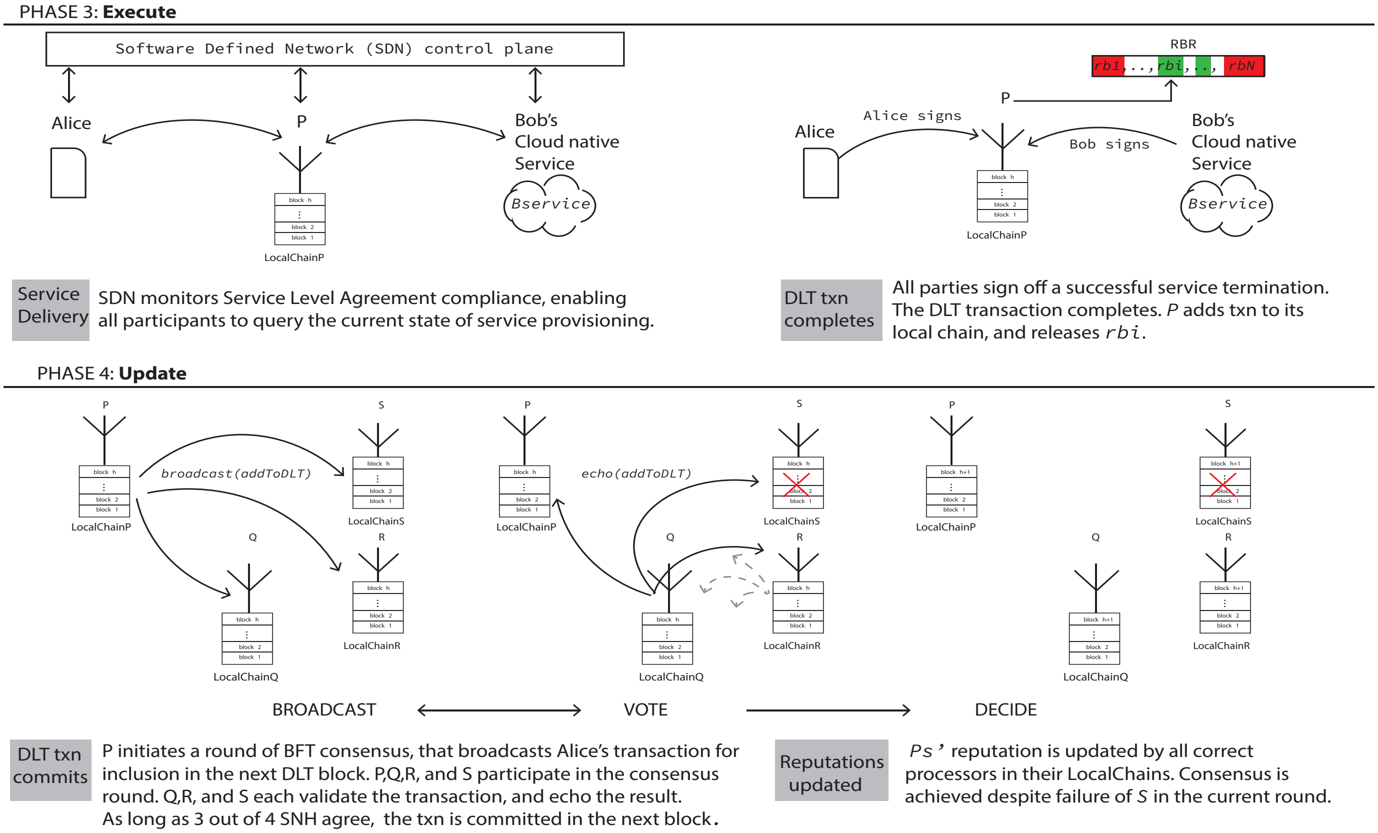
Alice selects a response: P, Q & R each respond with their Bservice price, and a list of requested reputations. Alice freely selects SNH *P*, and sends it proof of existence, and ownership, of funds (in Trust Coins).

PHASE2: **Setup**

SNH selects Alice's request: *P* accepts Alice's request among others if: Alice's proof of ownership validates against `LocalChainP`. *P* can acquire resource block (`rbi`) from local RBR.

Jointly composed DLT txn: Alice, Bob, and P, initiate a secret sharing protocol, to generate encryption keys. P composes a DLT txn, that pays for `rbi`, and P's DLT txn processing fee.

Fig. 2: Phases involved in fulfilling a user-request



PHASE 3: **Execute**

Service Delivery: SDN monitors Service Level Agreement compliance, enabling all participants to query the current state of service provisioning.

DLT txn completes: All parties sign off a successful service termination. The DLT transaction completes. *P* adds txn to its local chain, and releases `rbi`.

PHASE 4: **Update**

DLT txn commits: P initiates a round of BFT consensus, that broadcasts Alice's transaction for inclusion in the next DLT block. P,Q,R, and S participate in the consensus round. Q,R, and S each validate the transaction, and echo the result. As long as 3 out of 4 SNH agree, the txn is committed in the next block.

Reputations updated: *Ps'* reputation is updated by all correct processors in their LocalChains. Consensus is achieved despite failure of *S* in the current round.

Fig. 3: Phases involved in adding completed transactions to the ledger

Authorized licensed use limited to: University of Auckland. Downloaded on June 28,2022 at 03:48:26 UTC from IEEE Xplore. Restrictions apply.

Therefore trust has often been applied to energy or bandwidth constrained systems such as Wireless Sensor Networks. However, technological advances along with higher bandwidths accessible at mmWave and higher spectrum bands, enable applying both trust and cryptography based approaches to managing wireless communication systems. The key insight of our system lies in securing trust management processes through cryptography.

Our system absorbs trust interactions within DLT transactions. A transaction in the system encodes identities, obligations, and outcomes of these interactions. A trustor applies an application-specific trust function over some subset of transactions. The threshold for trust set by an agent is based on subjective factors, such as their experience with the system, and the importance they attach to the task at hand. Our system adapts the token of a traditional DLT called a TrustCoin (TC), to serve as the native cryptocurrency. Furthermore, by pricing network services based on the provider's reputation, a transaction in our system explicitly records reputation information.

B. Communication System

We assume, service discovery, and provisioning occurs over a Software Defined Networking (SDN) substrate [14] employing hardware virtualization techniques such as Cloud Native network Function (CNF) [3] to define, and compose network services. CNF based services can be viewed as a sequence of tethered functions, which take as input the service request, and produce the final desired output. Depending on granularity of control defined by the request, checkpoints can be established along multiple input/output interfaces along this sequence. These in turn may be collaboratively monitored by transaction participants using Threshold Encryption schemes as such Threshold Public Key Encryption [2]. Such schemes are employed by the BFT consensus protocol used by the system, therefore setting them up for this purpose does not add computational cost.

We assume the existence of a dynamic spectrum management policy. Following the principle of frequency reuse, we divide the network into disjoint clusters of SNH devices.With each such set served by a exclusive resource block register (RBR), as shown in Fig. 3, thus reducing radio spectrum management to a single cluster and its allocated RBR. We also make following modifications to DLT transaction setup, composition, and validation. Transaction setup includes a check of the associated RBR for resource block availability. The transaction proceeds only if this condition is met. The owner of the relevant resource block, e.g the cellular operator, then adds a new output to the proposed transaction, that pays them the fee associated with the use of the resource. This fee is payed regardless of the outcome of the underlying service being transacted. Therefore every proposed transaction is committed. Associating this cost with every proposed transaction, prevents Denial-Of-Service (DoS) type attacks.

C. DLT Framework

DLT systems may be described using a four-layer architecture comprising the user interface layer, application programming framework, compute layer, and consensus layer. UEs submit requests generated over the user interface to the application programming framework, which applies a semantic interpretation. This semantic description is validated by the compute layer. Successfully validated transactions are forwarded to the consensus layer which batch processes them for inclusion in the ledger.

Desirable features of DLT systems are derived from the mathematically provable security properties of cryptography protocols. Trust processes in our system are defined over such constructions, thereby extending their guarantees into the trust realm.

Each DLT node maintains an independent copy of a ledger of network activity. A consensus, or agreement protocol is a distributed mechanism for nodes to synchronize the states of their individual ledgers. A BFT [12] consensus protocol progresses despite a quorum of protocol participants displaying malicious behavior. Such protocols comprise a propose phase followed by an accept phase. During the propose phase, participating nodes invoke the Reliable Broad-

Authorized licensed use limited to: University of Auckland. Downloaded on June 28,2022 at 03:48:26 UTC from IEEE Xplore. Restrictions apply.

cast primitive to disseminate their proposals, which are subsequently voted on for inclusion in their local ledgers. Broadcast and voting schemes are implemented over quorums of correct processors using threshold cryptography [2] [10].

Each SNH device is incentivized for participating in a consensus round by receiving a quantity of TC proportional with both their long term reputation, and their participation in the current round. This is in line with each SNH device able to act purely as a service delivery node, consensus node or both.

In PoW style protocols, all the newly minted tokens are awarded to one miner, however the payoff structure of our protocol needs to reflect its collaborative nature. In our system, SNH devices compute each other's share of the block reward, based on the correctness of protocol messages generated by each participants, which also serves as a measure of reputation. Each participating SNH generates its own coinbase transaction [1]. Participating SNH undertake an additional round of messaging to generate agreement over the independently computed coinbase transaction. Therefore, trust computation, storage and update is folded into DLT processes and data structures.

IV. Motivating Example

As described graphically in Figure 2, Alice wants to avail of *Bservice* provisioned by Bob. Alice first sends a query to discover SNH devices within network range. SNH devices *P, Q, R, S* respond with their credentials, which include their reputation, and cost for delivering *Bservice*. The cost is stated in units of TC, while the reputation is a normalized, numerical score. Alice reviews the responses and finds *P*'s terms suitable, and signals its acceptance. Alternatively, Alice may decide that none of the SNH devices suit her current need, and can abort or postpone the request. We assume the low-bandwidth control signaling occurs over publicly licensed spectrum bands.

SNH *P* reviews Alice's user-request among the others it has received, and initiates a secret-sharing protocol [9] with Alice and *Bservice* to generate transaction specific keys. These secret shares are combined to generate a digital signature, to prove TC ownership, and collaboratively monitor progress. Software defined controls monitor the status of *Bservice*, and communicate it to transacting parties. All parties sign the successful completion of *Bservice*, and *P* receives its payment. *P* stores the completed transaction in its local buffer, scheduled for inclusion in the DL.

During a given round, each participating SNH runs one instance of the agreement protocol for every proposal. Participant *P* deems a round complete when two-thirds or more of these instances terminate.

V. System Validation

We validate the system by generating, and model checking a formal specification. The specification is a mathematically precise description of system behaviour. A model checker generates all possible system traces resulting from the specification to determine whether it violates any specified property.

We specify our system using Leslie Lamport's Temporal Logic of Actions (TLA) [15]. TLA has been developed for distributed and concurrent systems, and is able to pick up subtle bugs missed by traditional verification tools such as Monte-Carlo simulations, and unit testing. A state-machine abstraction of the system is described using set theory and first-order logic. A state is a unique assignment of values to system variables. An action enables transitioning to a new state by acting on state variables. The property that needs checking is described as Temporal Logic formula, and forms an invariant of the system i.e., a formula that must be satisfied at every state along all sequences of states. Our specification consists of three sets of functionally distinct processes, representing UE that propose values, SNH which validate these proposals, and those that commit them to the DLT ledger.

We model consensus with a Byzantine Version of the popular Paxos [8] protocol. The protocol is extended to account for tokenisation, and protection against double spending [1]. Our specification satisfies Safety, and Correctness properties [15], such as not allowing double

Authorized licensed use limited to: University of Auckland. Downloaded on June 28,2022 at 03:48:26 UTC from IEEE Xplore. Restrictions apply.

spending of TC, and ensuring the local Unspent Transaction Output [1] of all correct processors are consistently edited.

VI. Open Issues and Challenges

Our proposed system needs to coordinate its actions across disparate components of a radio communication system. The deployed dynamics of radio resource management, and software defined controls, may necessitate tuning specified system parameters. Such as setting block, and transaction sizes based on network throughput and latency. Reconfiguration of the consensus protocol to manage changes to membership, without disrupting availability of the system, remains an open challenge.

Even though model checking our system specification validates its correctness, it is necessarily carried out over a finite model size. Therefore the impact of potentially limitless scaling remains unclear, especially given the well-known inability of BFT consensus protocols to scale in the presence of faults. This is mitigated by leveraging the frequency dependent partitioning of radio systems, to create clusters of consensus groups.

VII. Conclusion

This paper introduced a trust based framework to implement a massively shared, multi-tenant wireless communication system. Our system overcomes the challenges of deploying ultra-dense networks, by dramatically expanding the class of operator admitted into the system. We replace rigid trust, with flexible terms negotiated independently between transacting UE and SNH. Trust in our system is derived from the actual record of network activity enshrined as DLT transactions. Therefore trust functions may be freely defined over arbitrary subsets of transactions. We have outlined the operational, economic, and technical components of the framework and validated its safety properties. This work presents a new paradigm for pervasive radio communication systems of the future. However, it requires a sea-change in the way networks are built, operated and viewed by traditional carriers.

References

[1] A. M. Antonopoulos, *Mastering Bitcoin: Programming the Open Blockchain*. Sebastopol, CA: O'Reilly Media, 2017.

[2] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of BFT protocols," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna, Austria, Oct. 2016, pp. 31–42.

[3] S. D. A. Shah, M. A. Gregory, and S. Li, "Cloud-native network slicing using software defined networking based multi-access edge computing: A survey," *IEEE Access*, vol. 9, pp. 10 903–10 924, 2021.

[4] R. Jurdi, A. K. Gupta, J. G. Andrews, and R. W. Heath, "Modeling infrastructure sharing in mmwave networks with shared spectrum licenses," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 328–343, 2018.

[5] H. Yang, Y. Liang, J. Yuan, Q. Yao, A. Yu, and J. Zhang, "Distributed blockchain-based trusted multidomain collaboration for mobile edge computing in 5g and beyond," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 7094–7104, 2020.

[6] H. Yang, B. Bao, C. Li, Q. Yao, A. Yu, J. Zhang, and Y. Ji, "Blockchain-enabled tripartite anonymous identification trusted service provisioning in industrial iot," *IEEE Internet of Things Journal*, 2021.

[7] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *Journal of the ACM (JACM)*, vol. 27, no. 2, pp. 228–234, 1980.

[8] L. Lamport, "The part-time parliament," *ACM Transactions on Computer Systems*, vol. 16, no. 2, pp. 133–169, 1998.

[9] D. Boneh, M. Drijvers, and G. Neven, "Compact multi-signatures for smaller blockchains," in *International Conference on the Theory and Application of Cryptology and Information Security*, Brisbane, Australia, Dec. 2018, pp. 435–464.

[10] C. Cachin, K. Kursawe, A. Lysyanskaya, and R. Strobl, "Asynchronous verifiable secret sharing and proactive cryptosystems," in *Proceedings of the 9th ACM conference on Computer and communications security*, Washington, DC, USA, Nov. 2002, pp. 88–97.

[11] S. Nakamoto. (2008, Oct.) Bitcoin: A peer-to-peer electronic cash system. cryptography mailing list. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[12] G. Bracha, "Asynchronous byzantine agreement protocols," *Information and Computation*, vol. 72, pp. 130–143, Jan. 1987.

[13] M. Ben-Or, B. Kelmer, and T. Rabin, "Asynchronous secure computations with optimal resilience," in *Proceedings of the thirteenth annual ACM symposium on Principles of distributed computing*, New York, NY, Aug. 1994, pp. 183–192.

[14] ITUR-WP5D, "Minimum requirements related to technical performance for IMT-2020 radio interface(s)," ITU, Tech. Rep., 2017.

[15] K. Chaudhuri, D. Doligez, L. Lamport, and S. Merz, "Verifying safety properties with the tla+ proof system," in *Proceedings of the 5th International Conference on Automated Reasoning*, ser. IJCAR'10. Berlin, Heidelberg: Springer-Verlag, 2010, p. 142–148.

Dev P. Singh (M'19) is currently pursuing a Doctorate degree with the Department of Electrical, Computer, and Software Engineering (ECSE) at the University of Auckland.

Kevin W. Sowerby (SM'03) Professor Sowerby currently serves as Head of ECSE at the University of Auckland.

Andrew C. M. Austin (M'10) is currently a senior lecturer in the ECSE Department at the University of Auckland.

Authorized licensed use limited to: University of Auckland. Downloaded on June 28,2022 at 03:48:26 UTC from IEEE Xplore. Restrictions apply.