Byzantine Fault Tolerance Models for Distributed Coordination in Dynamic Spectrum Sharing

Amy Babay*, Prashant Krishnamurthy*, Ilia Murtazashvili◇, Xiaoxuan Qin*
*Informatics and Networked Systems
◇Public and International Affairs & Center for Governance and Markets
University of Pittsburgh

August 2025

Abstract

As demand for wireless services intensifies, dynamic spectrum sharing has become a critical challenge for spectrum governance. Byzantine Fault Tolerant (BFT) protocols offer a technically robust approach to decentralized coordination, enabling multiple users to make consistent spectrum access decisions even in the presence of faults, conflicting incentives, or adversarial behavior. Yet, the success of BFT systems depends not only on algorithmic guarantees but also on the institutional environment in which they operate. This paper unfolds the various dimensions of BFT - the organizational dimension, distributed systems, fault models and performance metrics in the context of spectrum sharing. We conceptualize spectrum as a modular bundle of rights and elucidate several BFT-based architectures that vary in how they structure coordination and fault tolerance. We conduct a basic simulation of a two-tier sharing of spectrum slices, and our simulation results highlight trade-offs in sensor deployment and organizational trust. We argue that decentralized spectrum governance structures may enhance the feasibility and legitimacy of BFT-based spectrum sharing as there are strong similarities to polycentricity. BFT thus represents more than a technical solution: it is an institutional testbed for scalable, adaptive, and rule-based spectrum governance.

1 Introduction

With the scarcity of spectrum bands that are not already allocated and/or assigned, dynamic spectrum sharing is becoming imperative for future communications, sensing, and other applications such as weather radiometry and radio astronomy. The National Spectrum Strategy Research and Development Plan [1] defines dynamic spectrum sharing as:

> ⋯ adaptive coexistence using techniques that enable multiple electromagnetic spectrum users to operate on the same frequencies in the same geographic area without causing harmful interference to other users (in cases where such users have an expectation of protection from harmful interference) by using capabilities that can adjust and optimize electromagnetic spectrum usage in real time or near-real time, consistent with defined regulations and policies for a particular spectrum band.

It is common in the current regulatory framework to have **allocations** made to specific applications in specific bands (e.g., fixed satellite communications - from a fixed ground station to satellites in the X-band from 7.25 - 8.44 GHz). In these bands, **assignments** are made based on a licensing mechanism to specific "users" which may be companies or communities. Such users are called **primary** users. Sometimes, the same spectrum bands are allocated for multiple applications, and the assignments are made on a **co-primary** basis. And finally, some allocations are made to applications on a **secondary** basis, where users should expect interference from the primary users. If users have already been assigned to specific bands, we refer to them as **incumbents**. **New entrants** are users seeking access to spectrum that has already been allocated and assigned and are typically, but not necessarily, secondary users.

Practical approaches for sharing spectrum have been investigated in recent years. In Citizens Broadband Radio Service (CBRS) in 3.55-3.7 GHz, one approach uses environmental sensors deployed by trusted third parties in conjunction with a Spectrum Access System (SAS) database to detect incumbent transmissions from US Navy radars and inform secondary users. A dynamic protection area, typically an exclusion zone, is used to evacuate transmissions from secondary CBRS devices in the area when such transmissions are detected. A second approach allows the incumbent to inform the database or secondary users of their transmissions. A third decentralized approach used in the Automatic Frequency Coordination (AFC) in the 6 GHz band allows nodes to locally make decisions about transmissions using location information of existing fixed links and propagation models. Incumbents are expected to have priority and excluding "rights" in most cases.

We expect future dynamic spectrum sharing regimes to be implemented (preferably in a distributed manner) to support efficient coordination of spectrum rights between the *multiple spectrum users* in the definition from the National Spectrum Strategy Research and Development Plan. We use the term "spectrum grant" in the paper to distinguish the temporary apportioning of spectrum to a node or user from the regulatory allocation and assignments.

A centralized authority like the SAS could "grant" spectrum slices and transmit powers in time and space to nodes based on rights and interference. But centralization has disadvantages: bottlenecks, delays, vulnerability to failures, and accidental or intentional errors in spectrum grants since decisions still depend on distributed information gathering (environmental sensors, self-reporting by nodes etc.). Distributed coordination can alleviate some of these challenges. Yet, distributed coordination between nodes deployed by potentially competing entities with different rights, having differing observations of interference, or experiencing failures (node crashes, measurement errors, message loss, or even security breaches leading to node compromises) is challenging.

The Byzantine fault model provides a promising approach to reason about such challenges: in this model, a fraction of nodes may behave in arbitrary or even malicious ways. Byzantine Fault Tolerant (BFT) protocols [2] enable correct nodes to make consistent decisions despite arbitrary behavior from a subset of nodes. In the spectrum sharing problem, BFT protocols can provide assurance that dynamic spectrum grants, over shorter durations than static allocations, are made in a safe manner (avoiding harmful interference) despite failures or malicious activity that cannot be tolerated with a single centralized authority. Further, while BFT protocols offer technical guarantees for consistency and fault tolerance, they also allow for decentralization in an organizational

sense. In the case of spectrum sharing, it is important to examine how authority and legitimacy are embedded in surrounding institutions – a theme that parallels earlier critiques of spectrum governance's centralized inefficiencies [3, 4]. In this paper, we also offer an institutional perspective to complement technical approaches to BFT. As we examine BFT approaches, we consider to a limited extent how the governance of radio spectrum hinges on defining, enforcing and coordinating access and use rights[1]. By analyzing spectrum sharing through property rights lenses, we include (to an extent) the institutional design factors that underlie emerging models of dynamic spectrum coordination. These institutional concerns echo longstanding debates about rights fragmentation and inefficient allocation in spectrum policy [7].

The paper is organized as follows. In Section 2, we provide an overview of BFT and spectrum property rights in the context of a range of current and proposed spectrum sharing regimes —- including the UK's short-term 2.3 GHz licenses, the U.S. CBRS system, unlicensed U-NII bands, hybrid sharing in the upper 6 GHz band, and coordinated access at 37 GHz. We then introduce decentralized spectrum coordination using BFT systems in Section 3, which is novel in that to our knowledge, this is the first work that explores the various dimensions therein. Drawing on insights from distributed computing, we explore how BFT mechanisms can be used to implement spectrum sharing in a two-tier system. We categorize the components of a BFT-based sharing system, model user and server interactions, and identify organizational and technical considerations for deployment. By comparing these arrangements, we show how BFT models offer a promising framework for future spectrum governance. We believe this aligns with Ostromian principles of polycentricity while addressing the challenges of real-time interference management and adversarial behavior.

2 Background

In this section, we first describe what spectrum property rights mean in Section 2.1 followed by Section 2.2 where we provide several examples of bands where spectrum sharing coordination (now and in the future) would need fault tolerance and security. We describe what BFT means in Section 2.3 and give an overview of BFT models and mechanisms that can be applied to spectrum sharing.

2.1 Spectrum Property Rights

Understanding how spectrum is shared, managed, and contested requires more than technical engineering or regulatory detail – it requires an institutional framework that explains how rights are defined, distributed, and enforced. In this section, we describe a property rights approach to

[1]Traditional licensing regimes treat spectrum as a form of private property, allocated through formal legal instruments such as auctions or administrative grants. This Coasean model has proven effective in stable environments with well-defined incumbents but often struggles to accommodate the dynamic, short-term, and context-sensitive demands of modern wireless systems. Experimental and legal analysis has shown how Coasean clarity transformed thinking about radio spectrum allocation, including its disruptive effects on legacy licensing regimes [5]. In contrast, the Ostrom–Schlager framework conceptualizes property as a modular bundle of rights—access, withdrawal, management, exclusion, and alienation—that can be distributed across actors and enforced through formal or informal institutions [6]. This approach recognizes the feasibility of shared and polycentric governance, even in the absence of full legal ownership.

provide that analytical foundation. Rather than treating spectrum as a physical commodity, we analyze it as a resource governed by bundles of rights, such as access, use, exclusion, and transfer, that are allocated across users and institutions. This perspective allows us to make sense of the tradeoffs involved in different spectrum-sharing regimes and to compare governance models across contexts.

We focus on two foundational perspectives: the Coasean view, which emphasizes formal, transferable rights enforced by central authorities; and the Ostrom–Schlager typology, which identifies a flexible bundle of rights that can be distributed across actors and managed in decentralized systems. These frameworks allow us to evaluate how spectrum can be governed through centralized, polycentric, or distributed institutions. This theoretical grounding is essential for understanding new technical proposals such as Byzantine Fault Tolerant (BFT) coordination systems. While BFT is a promising mechanism for enabling decentralized spectrum management, its practical operation depends on the surrounding institutional environment: how rights are defined, who can enforce them, and what forms of authority or legitimacy exist. As explained later, servers that participate in a BFT protocol may be deployed by multiple organizations. What rights they hold over spectrum slices may determine their roles in a BFT protocol. For example, if some hold primary rights in in one band while others hold primary rights in a different band and the organizations would like to coordinate, the implementation of a BFT protocol may be different[2]. A deep understanding of that institutional context requires a coherent analytical framework. The property rights tradition provides such a framework. The sections that follow apply this framework to several prominent spectrum-sharing arrangements, setting the stage for a future evaluation of where and how BFT systems may be a viable complement or alternative.

The property rights approach disaggregates control over a resource into distinct functions, such as access, use, exclusion, management, and transfer, and analyzes how those rights are allocated and enforced [8, 9]. Rather than assuming a single form of ownership or control, this approach allows us to describe and compare governance systems in terms of the specific rights they confer on different actors.These foundational ideas have informed critiques of top-down allocation [3, 4] as well as proposals for more self-organized rights regimes [10].

We focus on two complementary frameworks. The Coasean perspective emphasizes exclusive rights defined and enforced through legal mechanisms and markets. The Ostrom–Schlager typology sees property as a modular bundle of rights that can be shared, partially held, or enforced through decentralized institutions. Together, these frameworks provide a powerful lens to analyze centralized, polycentric, and distributed models of spectrum governance.

2.1.1 The Coasean Perspective: Property through Formalization and Transfer

The Coasean framework treats property rights as tools to internalize externalities and facilitate voluntary exchange. In spectrum policy, this has traditionally meant assigning clearly defined licenses—exclusive rights to transmit over a geographic area or frequency range—via auction or administrative decision. The Coasean approach is suitable for static allocation or at best a centralized allocation of spectrum in a sharing scenario (which is similar to cellular service providers who support their mobile users). This model depends on a central legal authority to define, as-

[2]In this paper we assume a single band and a two tier system, but this work can be generalized.

sign, and enforce rights, making it highly effective in stable environments with large incumbent users (as with the cellular service providers). However, as spectrum use becomes more dynamic, real-time, and context-dependent, the Coasean model shows its limits. Rigid licenses often fail to accommodate short-term or location-sensitive applications, prompting interest in more flexible, decentralized alternatives.

2.1.2 The Ostrom–Schlager Typology: Bundled Rights and Polycentric Management

Elinor Ostrom and colleagues introduced a framework better suited to common-pool resources, where access is overlapping and exclusivity is difficult. They distinguish among five types of rights:

- **Access** – the right to enter or monitor the resource
- **Withdrawal** – the right to extract or use resource units (e.g., spectrum airtime)
- **Management** – the authority to set or modify internal rules
- **Exclusion** – the authority to determine who may access the resource
- **Alienation** – the right to sell or lease any of the above rights

These rights can be held in combinations and need not all be present for a governance system to function. In many cases, users or communities govern effectively with only a subset of rights and without formal legal ownership. For instance, community networks may exercise management and exclusion rights through technical means or social norms, even in unlicensed bands. This bundle-based approach to governance has been explored further in Ostrom's later work on institutional diversity [11].

Importantly, Ostromian governance is not synonymous with anarchy: it assumes some institutional framework, even if not centralized. Enforcement can come from local governments, regional bodies, or distributed technical protocols. Recent research on the amateur radio community illustrates how polycentric governance can function without centralized control, relying instead on norms, reciprocity, and layered institutions [10]. The framework is also agnostic to formality – a government may restrict exclusion rights even in licensed regimes, and informal systems may enforce usage rights through repeated interaction, reputation, or decentralized consensus (as with BFT systems).

2.2 Spectrum Sharing

We can now analyze contemporary sharing regimes across several key bands. These examples show how property rights—defined as bundles of access, exclusion, and management rights—are structured under different technical and institutional arrangements. Table 1 summarizes several key bands, their incumbents, new applications, and the mechanisms used for sharing. These range from centralized licensing and automated coordination to open-access regimes with weak or post hoc enforcement. Each provides insight into how spectrum property rights are allocated and contested in practice. These cases exemplify a broader shift toward hybrid and polycentric spectrum governance, challenging the assumptions of purely centralized models [12, 13].

Band	Incumbents	New Application	Potential or Current Sharing Mechanism
2.32-2.34 MHz	Amateur radio, Low-power indoor, military; emergency services (adjacent channel)	Events, News, 5G cameras	Short duration (14 days), pre-planned exclusion zones, fixed separation distances
CBRS – 3.55-3.7 GHz	US Navy Radar, fixed satellite links	Private and public networks, mostly 4G, some 5G and other technologies	Using a SAS with environmental sensors and incumbent informing capability
5 GHz (U-NII 2a and 2c)	Terminal Doppler Weather Radar (FAA)	Wi-Fi	Dynamic Frequency Selection (DFS)
Upper 6 GHz	Fixed links, satellite links, broadcast auxiliary service	WiFi (unlicensed) or cellular (licensed)	Using automated frequency coordination; also hybrid sharing between cellular and WiFi
37 GHz	Some federal, but mostly empty	point-to-point or base station to devices links	First come first serve with subsequent coordination through the FCC

Table 1: Example bands where spectrum is shared and has the potential for a BFT system

2.3 GHz short-term sharing in UK: In July 2025, Ofcom announced a consultation called "Enabling short notice, short duration licenses in 2.3 GHz" where temporary licenses were proposed in the 2.32-2.34 MHz band. The expectation was that 10-20 MHz of spectrum here could be used for short-term applications like breaking news, events (sports, concerts) and even private network demonstration [14]. In this context, a BFT system could be used to take requests and allocate resources that would avoid overlaps in space and frequency.

CBRS in 3.55-3.7 GHz bands: The Citizens Broadband Radio Service (CBRS) [15] in the U.S. is a hallmark example of a tiered spectrum sharing regime. Operated under FCC rules, it segments users into three access tiers: Incumbent Access (e.g., U.S. Navy radar), Priority Access License (PAL) holders, and General Authorized Access (GAA) users. A dynamic Spectrum Access System (SAS) manages these tiers in real time, ensuring incumbents are protected while optimizing spectrum use among commercial operators. PAL licenses are auctioned and provide semi-exclusive rights over localized areas, while GAA permits opportunistic, unlicensed-like access, but has no protection from interference. The CBRS model exemplifies how algorithmic coordination can support efficient coexistence without fully displacing legacy users. In this paper, we consider a two-tier system similar to CBRS, where Type 1 users (like CBRS incumbents) have priority over Type 2 users. We do not explicitly consider PAL or GAA in our models.

5.25-5.35 GHz and 5.47-5.725 GHz U-NII bands: These bands are part of the broader U-NII allocation supporting Wi-Fi and other unlicensed uses. Sharing in these mid-band frequencies is governed through dynamic frequency selection (DFS) mechanisms designed to avoid interference with federal radar and satellite systems. Devices must detect incumbent signals and automatically

vacate channels to prevent disruption. U-NII sharing represents a relatively decentralized sharing model, relying on device-level sensing and compliance by vendors. The latter (compliance) is harder to achieve, and BFT protocols could be employed in such a case to coordinate between devices.

Hybrid sharing in the upper 6 GHz bands In the upper 6 GHz band (6.425–7.125 GHz), the FCC authorized standard power unlicensed use with automated frequency coordination (AFC), alongside lower-power indoor use without coordination. This hybrid approach allows coexistence between unlicensed users and incumbent fixed-service microwave links. AFC systems operate similarly to SAS in CBRS or the DFS mechanism using location information rather than signal detection, but are tailored to static incumbent deployments rather than mobile or real-time coordination. This model illustrates a compromise between the openness of unlicensed spectrum and the reliability needs of incumbents, enabled through cloud-based registries and interference calculations.

The 37 GHz Coordination: The 37–37.6 GHz band is a novel case of coordinated sharing between non-federal and federal users. Instead of traditional licensing, the FCC designated this segment for coordinated co-equal access, where commercial users must coordinate their operations through a shared database and negotiation framework. Its governance regime reflects a polycentric model (with the database requirement) of sharing—more flexibly than rigid auctions, but still structured to prevent harmful interference. It represents an experiment in non-exclusive access with structured self-governance mechanisms. This intellectual trajectory can be traced to early arguments for market-based allocation, such as Herzel's 1951 proposal later credited by Coase himself **[16]**. BFT would be a good option for such coordination, taking into account the challenges of a centralized database.

2.3 Byzantine Fault Tolerance (BFT)

The previous examples illustrate a range of spectrum sharing arrangements that blend centralized oversight with varying degrees of decentralized coordination—whether through short-term licensing, tiered access systems like CBRS, or automated frequency coordination in the upper 6 GHz and 37 GHz bands. These models depend on accurate, timely information and trusted mechanisms to allocate access while minimizing interference. However, as spectrum environments become more dynamic and heterogeneous, especially with the proliferation of users and devices in dense or contested settings, existing coordination mechanisms may struggle with reliability, trust, or scalability. To address these challenges, researchers have proposed adopting Byzantine Fault Tolerant (BFT) models—borrowed from distributed computing—to support resilient, decentralized spectrum sharing systems. BFT approaches offer a framework for ensuring agreement and consistent operation among a network of servers or agents, even when some may fail or behave maliciously. The following section outlines key building blocks of BFT-based spectrum sharing, including system and fault models, coordination mechanisms, and practical fault tolerance strategies, as well as related work on BFT in the spectrum sharing context.

Byzantine Fault Model: The Byzantine fault model was introduced in the context of the Byzantine Generals Problem [17]. Under this model, a threshold number of participants (e.g. servers) may be *Byzantine*, meaning they can behave completely arbitrarily, including maliciously. Byzantine participants can deviate arbitrarily from the protocol followed by correct participants: for example, they may fail to perform the actions specified by the protocol or fail to send messages to other participants, or may send messages that are corrupted or maliciously crafted to attempt to subvert the protocol and cause disagreement among correct participants or prevent them from reaching a decision. The number of tolerated Byzantine participants is typically denoted by f. All other participants are assumed to be *correct*.

The Byzantine fault model has been extensively studied in the distributed computing literature, in settings that vary in their assumptions regarding *timing* (also referred to as *synchrony*) and *cryptography*. For additional discussion of these models, see, for example [18, 19, 20, 21]; here we provide a brief overview. In the *synchronous* setting, there is assumed to be a known fixed bound on all message and computation delays, such that participants can execute the protocol in lockstep rounds, where each participant is guaranteed to receive all messages sent to it by other participants in a given round by the end of that round. However, this model is not realistic in practical networks where messages may be delayed or lost. In the *asynchronous* model, there are no bounds on how long it may take messages to arrive. Unfortunately, the agreement problems we are interested in are impossible to solve deterministically in an asynchronous model [22]. Thus, practical protocols must either be randomized (and only provide probabilistic guarantees) or, more typically, assume an intermediate *partial synchrony* timing model, which assumes that message delays may initially be unbounded but are *eventually* upper bounded by some constant after some unknown *global stabilization time*. In practice, protocols in the partially synchronous model typically guarantee *safety* (Byzantine processes cannot cause correct processes to reach conflicting decisions) in all cases but only guarantee *liveness* (correct processes eventually reach a decision) during periods of synchrony, when the network is sufficiently stable (delays are bounded).

All of the BFT protocols we consider assume that a participant that receives a message knows which participant sent that message; this can be achieved either through a physical network setup with direct point-to-point links or (typically more practically) through point-to-point cryptographic message authentication. Additionally, protocols may assume that messages can be *digitally signed* such that a participant that receives a message can forward it to other processes who can verify that the message was in fact sent by the original participant that created it (and the message cannot be modified by the forwarding participant).

We assume the spectrum sharing scenario we consider will typically operate in a partially synchronous setting where digital signatures are possible. In this setting, protocols solving the agreement problems we discuss below typically requires that the total number of participants is at least $3f + 1$ in order to tolerate f Byzantine participants (equivalently, number of Byzantine participants must be strictly less than one-third of the total number of participants).

BFT Agreement, Consensus, and Interactive Consistency: The literature on Byzantine Fault Tolerance provides a framework for how to *tolerate* Byzantine faults, i.e., to enable correct participants to reach consistent decisions and allow the system as a whole to operate correctly despite Byzantine behavior from a subset of participants. There has been extensive work on Byzantine

agreement problems, where the goal is for all correct processes to decide on a value. The basic requirements are that all correct processes decide on *the same* value (agreement) and all correct processes eventually decide (termination, or liveness). There are many variants of this problem that differ in the specific system setup and in the specific nature of and requirements on the value that is decided (validity). For simplicity, here we focus on the basic definitions of Byzantine *agreement*, *consensus*, and *interactive consistency* as specified in [18] (note that we refer to all three of these problems collectively as Byzantine agreement problems). We give informal descriptions of these problems below. See [18, 19] for additional, more formal description, and [23] for more recent work on validity property variants.

- Byzantine Agreement: One participant proposes a value. If the proposing participant is correct, all correct participants must decide on the proposed value. Otherwise, all correct participants must decide on the same value (but it can be any value, as long as it is the same for all correct participants).
- Consensus: Each participant has an initial value, and each participant proposes its initial value. If all correct participants have the same initial value, they must all decide on that value. Otherwise, all correct participants must decide on the same value (but it can be any value, as long as it is the same for all correct participants).
- Interactive Consistency: Each participant has an initial value, and each participant proposes its initial value. All correct participants must decide on the same *vector* of values $(v_1, v_2, \ldots, v_n)$, where entry v_i corresponds to the value of participant i. If participant i is correct and has initial value v_i, then all correct participants must decide on v_i as the i^{th} vector entry. If participant i is not correct, correct participants may decide on any value for the i^{th} vector entry (but must all decide the same value).

As observed in [18, 19], these problems are closely related to each other: for example, any protocol for Byzantine Agreement can be used to provide Interactive Consistency by simply running n parallel instances of the Byzantine Agreement protocol (one per participant). In Section 3, we discuss how the problem of dynamic decentralized spectrum sharing can be mapped to variants of Byzantine agreement problems.

BFT State Machine Replication (SMR): Beyond the basic agreement problems discussed above, which focus on deciding on a single value (or vector of values), BFT State Machine Replication (SMR) is an important framework that we will consider for implementing BFT spectrum sharing systems. State Machine Replication is a fundamental technique that provides fault tolerance by replicating an application's state across a set of servers (replicas): replicas agree on the *order* in which to process requests, and by processing (deterministic) requests in the same order, all replicas progress through an identical sequence of states [24]. One way to view BFT SMR is as a sequence of consensus instances, where all replicas may propose operations to perform, and the first operation is the one decided in the first consensus instance, the second operation is the one decided in the second consensus instance, and so on. But, explicitly designing protocols for the SMR setting can provide more efficient solutions, with the first practical BFT SMR protocols designed by Castro and Liskov [2]. Since then, many BFT SMR additional protocols have been developed to improve

normal-case performance (e.g. [25, 26]), performance under attack (e.g. [27, 28, 29]), or scalability (e.g. [30, 31]).

Related Work on BFT and Spectrum Sharing: Prior work that relates to Byzantine Fault Tolerance and spectrum sharing primarily falls into two categories: (1) blockchain-based spectrum management systems and (2) spectrum sensing schemes that tolerate Byzantine participants.

Prior work has proposed that blockchains can be useful for spectrum management [32], which has led to a large number of blockchain-based spectrum management or spectrum trading systems being proposed [33]. Many (though not all) blockchains are designed to tolerate Byzantine behavior from a subset of participants. BFT SMR, as described above, is often used as the underlying technology to implement blockchains that tolerate Byzantine faults [34], particularly for *permissioned* blockchains that operate between a limited set of authorized participants: participants propose blocks of transactions, and a BFT SMR protocol is used to determine the order in which to process blocks (and append them to the blockchain). Most of the existing work in this space assumes that a blockchain infrastructure already exists and does not specify a clear fault/threat model or consider how the blockchain infrastructure needs to be configured in order to support that threat model. The most relevant works in this space are BD-SAS [35] and TrustSAS [36]. BD-SAS introduces a two-level blockchain, where a global chain is used for regulatory actions, and local chains are used for allocating spectrum to users within a particular geographic region. The model of dividing management into local chains is similar to our Regional Agreement model (Section 3.2.2). TrustSAS focuses on protecting the privacy of secondary (Type 2) users and employs a hierarchical approach where *clusters* of secondary users obtain spectrum allocations via a global blockchain, and then make individual spectrum grants within each cluster. In general, this line of work focuses on building spectrum allocation or trading systems for specific requirements on top of existing blockchain systems. In contrast, we focus on mapping the dynamic spectrum sharing problem to fundamental BFT mechanisms (which also underlie blockchain systems) and consider how different design choices can influence the specific fault tolerance properties, cost, and performance of a decentralized spectrum sharing system.

Byzantine behavior has also been considered in the context of spectrum sensing for cognitive radio [37]. In this case, the typical model is that there are a number of cognitive radios (CRs) that report sensing data to cooperatively determine whether or not a primary user is active, and each CR has some probability of being Byzantine. Existing work develops data fusion schemes that can (with high probability) accurately detect primary user presence/absence in this environment. This line of work generally focuses only sensing, not on a system for allocating spectrum based on sensing results. In contrast, we consider both sensing and allocation, as well as the organizational relationship between sensors and spectrum granting servers. In this paper, we focus on classic threshold-based BFT models, where we assume up to a specific number of sensors may be compromised. However, techniques from the CR spectrum sensing literature may be useful in extending the work to cover probabilistic sensor fault models. In this case, we could potentially apply the data fusion techniques from past work in the internal logic for how servers process sensor reports. One work on spectrum sensing with Byzantine nodes that is similar in spirit to our approach focuses on low-earth-orbit satellite constellations [38]. It maps the problem of different operators with noisy analog measurements reaching agreement on spectrum use in a given time

interval to exact and approximate variants of the Byzantine Agreement problem [38].

3 Spectrum Sharing and BFT

In this section, we describe the various facets that need to be considered in the design of BFT models for spectrum sharing. We categorize these into organizational aspects, system aspects (including consensus approaches), fault models that can cause challenges in spectrum sharing, and BFT performance metrics as shown in Figure 1.

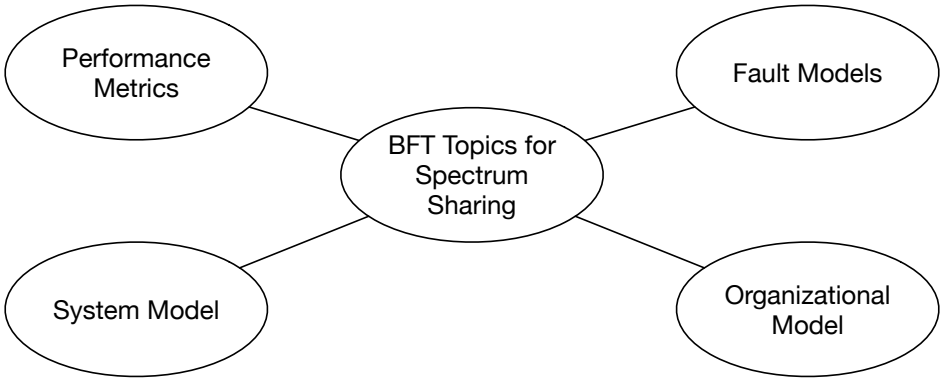


Figure 1: Overview of topics that need consideration in BFT Models

We conceptualize spectrum sharing as being accomplished by a set of environmental sensors and servers. The environmental sensors report what they measure (i.e. transmissions that they detect) to the servers. The servers assign spatial and temporal grants to spectrum slices to users making requests to these servers. To make assignment decisions, the servers maintain information regarding ongoing transmissions and current spectrum grants, essentially implementing a decentralized database of spectrum usage. We refer to the usage information stored at the servers as the *system state*. In what follows, we make the following assumptions:

- There are N_1 users of Type 1 that can be considered as either the primary users or incumbents, with priority over secondary or new entrant users of Type 2, of which there are N_2 in number. In the general case, there may be users of many types Type j where $j = 1, 2, \cdots \mathcal{N}$. For simplicity we assume $\mathcal{N} = 2$.
- There are S spectrum slices in a band that are available for sharing. In the more general case, there will be S_i spectrum slices in band B_i, $i = 1, 2, \cdots, \mathcal{B}$. To simplify discussion, we assume that we have one spectrum band with similar radio propagation characteristics unless otherwise discussed. In the case of multiple bands, each band might have its own propagation characteristics, and the bands need not be fungible [39].

Coordination among the servers is needed to ensure that they do not issue conflicting spectrum grants (e.g., allowing grants to Type 2 users that interfere with transmissions from Type 1 users, assigning the same spectrum slice to multiple Type 2 users, or allowing grants that cause interference between users). In this model, we do not explicitly consider the "rights" over spectrum

resources, except that Type 1 users have the highest rights. Type 2 users are treated homogeneously (i.e., they have the same rights and have agreed to negotiate for access grants through the system described next).

3.1 Organizational Considerations

There are K servers with associated databases that are responsible for making spectrum grants to Type 2 users. The K servers receive reports of detected Type 1 user transmissions from M environmental sensors deployed throughout the geographic area in which the spectrum sharing occurs. The organizational structure of who is responsible for deploying and managing these servers and sensors influences the trust assumptions underlying the system design, the threat/fault models the system must address, and feasibility constraints in the system design.

In the simplest case, the K servers and M sensors can all be deployed by a single organization. Such an approach may still be distributed (i.e., using redundant servers and sensors to tolerate failures and/or compromises) but is logically centralized and requires all users to fully trust the organization running the system (similar to CBRS today). However, we expect future spectrum sharing schemes to be decentralized, with multiple organizations each deploying a subset of the involved servers and sensors. As one example, a third party that both types of users of a particular band trust might deploy the infrastructure, or there may be multiple third parties competing to provide the service. In general, we can assume O organizations, where each organization O_i deploys k_i servers and m_i sensors. Some Type 1 and Type 2 users may also belong to the organizations deploying servers and sensors.

The organizational model influences our system model. In particular, to simplify system management, we may choose to have sensors only "know about" and send reports to servers managed by the same organization. This would limit the need for coordination when an organization deploys or decommissions servers or sensors (e.g. new sensors only need to be configured to know about servers of the same organization; a single organization replacing a server does not require updating all sensors at all organizations). In the polycentric spectrum rights model, the assumption is that all participants can monitor resource usage. We do not consider this explicitly, but data exchanged between servers deployed by different organizations could satisfy this requirement. Alternatively, sensors could report to all servers by "bootstrapping" using servers from their own organization to learn about the servers from other organizations.

The organizational model also affects our trust assumptions. We naturally expect there to be a higher degree of trust within an organization compared to across organizations (e.g. a server has reason to give higher weight to reports from its "own" sensors). This is especially true because organizations may be competing entities and may have incentives to prefer granting spectrum to users from their own organization if possible.

The trust aspect is closely related to the organizational influence on the fault/threat model. Classic BFT models typically consider each component separately and can tolerate a fixed fraction or threshold number of faulty components. This naturally captures failures or compromises of individual devices (i.e., servers or sensors), but in practice, we may also encounter faulty behavior at the organizational level. For example, an organization may implement intentional policies that prefer its own users, may suffer accidental misconfigurations that affect all of their sensors, or may be subject to an organizational-level compromise or insider threat that compromises all of

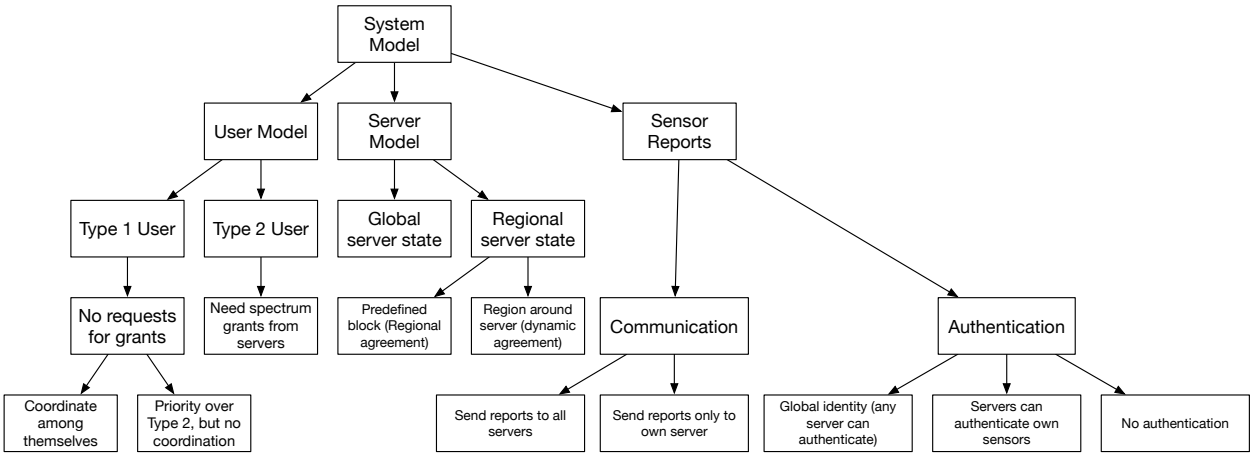


Figure 2: Outline of System Model

their servers.

3.2 System Model

The overall system model includes models for each of the three core components of the system (users, servers, and sensors) and how they interact with each other. Figure 2 shows the categorization of the system into user models, server state, and how sensors report. These are described in the following sections.

3.2.1 User Model

We assume Type 2 users make requests for spectrum grants and only transmit upon receiving a grant. Type 1 users do not make requests and may transmit at any time. Upon detecting a Type 1 transmission, servers should issue a revocation to any Type 2 users with active transmissions that may interfere with the detected Type 1 transmission. As an alternative to the explicit revocation model, we could consider a heartbeat model, where grants to Type 2 users are implicitly revoked if they do not receive a heartbeat from the servers for a specific amount of time. The heartbeat model could be implemented in multiple ways: (1) Type 2 users could be required to periodically refresh their grants by sending a new request to the servers after a certain period of time, or (2) servers could track all "live" grants and send periodic heartbeats inform Type 2 users that their grant is still valid. The heartbeat model could be combined with the explicit revocation model, such that grants are implicitly revoked if heartbeats are not received but may also be explicitly revoked prior to the heartbeat expiration.

Because Type 1 users do not need to wait for a grant before transmitting, there may be a short period of interference at the beginning of a Type 1 transmission before it can be detected and the interfering Type 2 user(s) evacuated.

The model where Type 1 users do not need to report their transmissions or make requests assumes that there is some form of out-of-band coordination among Type 1 users so that they do not interfere with each other. This coordination may be largely static (e.g. ensuring there is only a single primary user assigned to any particular slice) or may be more flexible. In general, we prefer

to avoid requiring Type 1 users to report their activities due to confidentiality concerns (e.g. from the US Navy as a current incumbent user). However, we could consider a two-stage grant process where Type 1 users report some (potentially obfuscated) information in a subsystem dedicated to agreement among Type 1 users only (and not publicly exposed to Type 2 users), and outputs from that subsystem are reported to the servers running the overall spectrum sharing system.

3.2.2 Server Model

We consider three possible models for servers to coordinate their decisions regarding spectrum grants.

Global Agreement. The Global Agreement model maps to classical BFT state machine replication protocols [2] and is the simplest of the three models. All servers maintain a complete view of the system state (i.e., current spectrum usage over the entire geographic area served), and all (correct) servers must agree on the decision for each client request. Avoiding conflicting allocations in the Global Agreement approach is simple, but scalability is a challenge. This model aligns with the Coasean view of a singular authority granting rights to spectrum slices (albeit in this case, only for a finite period of time). The rules governing spectrum grants are encoded in the deterministic logic that each server executes when processing a request in order to decide whether or not it should be granted; these rules are identical for all servers (including those deployed by different organizations).

In this model, servers run a single global BFT state machine replication (SMR) protocol. Each user request and each sensor report is forwarded to all of the servers, and they execute the BFT SMR protocol to agree on the order in which to process requests/reports. Each server processes every user request and sensor report in the order determined via the BFT SMR protocol. Since the processing of each request is deterministic, agreeing on the order in which to process requests guarantees that all (correct) servers will have the same view of the current state and make identical decisions when processing each request. A standard BFT SMR protocol tolerates a fixed number f of faulty servers and requires a total of $3f + 1$ servers in order to guarantee that all correct servers will agree on the order of operations (and thus on the state and spectrum grant decisions).

Specifically, when processing a user request, a server calculates whether the request will interfere with any ongoing transmission using a propagation model or other considerations. If the transmission will *not* cause interference, the server generates a *grant* response and updates its internal state (spectrum usage database) to include that transmission. Otherwise it generates a *reject* response. In order to tolerate potentially malicious server behavior, a single grant response is not enough for a user to begin transmitting. Under the assumption that up to f servers may be faulty, a user must wait for $f + 1$ grant responses to verify that at least one correct server agreed to grant the request. Upon collecting $f + 1$ reject responses, a user can conclude that the request was rejected.

Similarly, when processing a sensor report of a detected Type 1 transmission, a server calculates whether the reported transmission interferes with any ongoing Type 2 transmissions. If so, it generates an *evacuate* command for each interfering transmission. Upon receiving $f + 1$ evacuate commands, a Type 2 user must stop their transmission. Note that this description assumes that all sensor reports are trustworthy, since a server generates its evacuate response upon processing a single sensor report. To tolerate faulty sensors, a more involved decision procedure is needed. For

example, to tolerate a fixed number of faulty sensors g a server should only generate an evacuate command upon receiving $g + 1$ reports from different sensors indicating the same transmission information. However, this requires that sensors are deployed densely enough that $2g + 1$ sensors are in range of (able to detect) any transmission (so that $g + 1$ *correct* sensors are guaranteed to be available and able to detect the transmission). See Section 3.3 for additional discussion of faulty sensors.

Regional Agreement. The Regional Agreement model maps to sharded BFT SMR protocols [40, 41]. The general operational flow is similar to the Global Agreement model, but it breaks the geographic service area into regions, so that each spectrum allocation server is assigned to a region (e.g. the geographic region it is physically located in) and typically only needs to maintain sensor data for its own region and process requests from clients in this region; servers in the same region must agree on actions in that region.

However, we must also handle transmissions at the boundaries between regions, which may cause interference in multiple regions. In this case, a simple strategy for handling such transmissions is as follows. When a server processes a request, it calculates whether the request will interfere with any ongoing transmission using the same method as in the Global Agreement model, but based only on knowledge of transmissions within its own region). If so, the server can immediately generate a reject response. If there is no interference within the region, but the transmission's interference zone extends beyond the boundaries of its region, it provisionally updates its own state to include the transmission and forwards the request to the other affected region(s). The servers in the other regions treat the request the same as any other incoming request: they use their BFT SMR instance to order it, and when processing it they determine whether it interferes with any ongoing transmission. If it interferes, they send a reject response to all servers in the "home" region for the client. If it does not interfere, they provisionally update their state to include the transmission and send a grant response to all servers in the "home" region. Upon collecting $f + 1$ grant requests from all other affected regions, a server in the home region generates a grant response for the client. If a server receives $f + 1$ reject responses from any region, the server updates its state to remove the transmission and generates a reject response for the client. It also sends its reject response to the servers in the other regions to allow them to update their own states to remove the request (upon getting $f + 1$ reject responses from the home region).

For sensor reports, upon processing a report, if its interference zone extends beyond the region, a server forwards the report to the other affected region(s). Each affected region runs the BFT SMR algorithm to order the request and generates evacuate commands for affected Type 2 users within their region in exactly the same way as if the report originated from a sensor in its own region. Such collaborative coordination is an example of polycentricity, where different regions do not operate in silos, but work together using the information they have gathered. The collaboration could extend to multiple spectrum slices, their rights, and temporal grants.

More efficient technical solutions to cross-region coordination are likely possible, drawing on the existing literature on cross-shard transactions [41]. One simple approach to improve latency is to optimistically forward requests/reports that cross regions to the relevant regions *before* ordering, so that the relevant regions can all order the request/response in parallel. However, this has the drawback of generating more cross-region traffic, since some requests will be forwarded that would be rejected based only on the result from their "home" region. This creates a tradeoff

between throughput and latency that needs to be assessed based on the expected request characteristics for a given deployment.

Note that this model requires deploying $3f + 1$ servers per region in order to guarantee that it can tolerate any f faults. Therefore, one question to investigate is how to best divide the geographic space into regions and assign servers to regions. We note that a single physical server may participate in multiple BFT SMR instances (i.e. for multiple regions). In the extreme, we could even have all servers participate in all regions. While this may initially seem equivalent to the Global Agreement model, with a single set of servers processing all requests, it may still have performance benefits by allowing requests to be processed in parallel rather than forcing a total ordering over all requests. However, greater latency benefits are likely to come from dividing servers into geographic regions, such that the servers participating in a given BFT SMR instance are all physically close to one another and to the clients they serve.

Dynamic Agreement. Dynamic Agreement is a novel model that aims to minimize coordination overhead while still preventing conflicting allocations. Regions are not predetermined; instead each server maintains sensor data for its own local region, and the set of servers that must agree on the outcome for each client request is determined dynamically based on the client's location and request characteristics. In this model, each server has a geographic zone that it is responsible for; typically, this will be an area of a given diameter centered around that server's physical geographic location. Responsibility zones for different servers can overlap. In fact, for fault tolerance, these zones must overlap. To tolerate f Byzantine servers, we require that every point in the geographic space is covered by $3f + 1$ servers.

When a Type 2 user wants to transmit, they calculate the "interference zone" for their request and contact all servers responsible for that area. Note that this increases the client-side logic requires, since the client needs to determine who to contact. Alternatively, a client could be configured to know about any set of servers, and servers could be responsible for forwarding requests to the correct servers for the relevant zone. For handling requests, classic BFT SMR does not make sense here, as it is not possible to agree on a total ordering of requests if the set of relevant servers may be different for each request. Instead, we propose to run a BFT *interactive consistency* instance on the result of each request. That is, for each request, each server in the relevant zone checks if request is should be granted or rejected based on its own state. If a server believes the request should be granted, it tentatively reserves the requested spectrum. Then, the servers run a BFT interactive consistency protocol, where each server proposes its grant/reject vote as its value. The interactive consistency protocol guarantees that all correct servers agree on the vector of grant/reject votes. Then, each server can locally decide whether to issue a grant or reject response based on the vector contents. Specifically, if the vector contains at least $f + 1$ reject votes, servers send a reject response to the client (and release the tentative reservation if they had made one); otherwise, they send a grant response. To prevent interference, we want to reject a request if any correct server indicates that it will cause interference. However, we do not want Byzantine servers to be able to maliciously deny requests that would not cause any interference. Thus, we use the threshold of $f + 1$ so that f Byzantine servers cannot unilaterally reject requests. To ensure that $f + 1$ correct servers will correctly reject any response that would cause interference, the system must be configured such that $2f + 1$ servers are aware of any potentially interfering transmission. Dynamic agreement is another example of polycentricity where individual servers and their sensors, rather

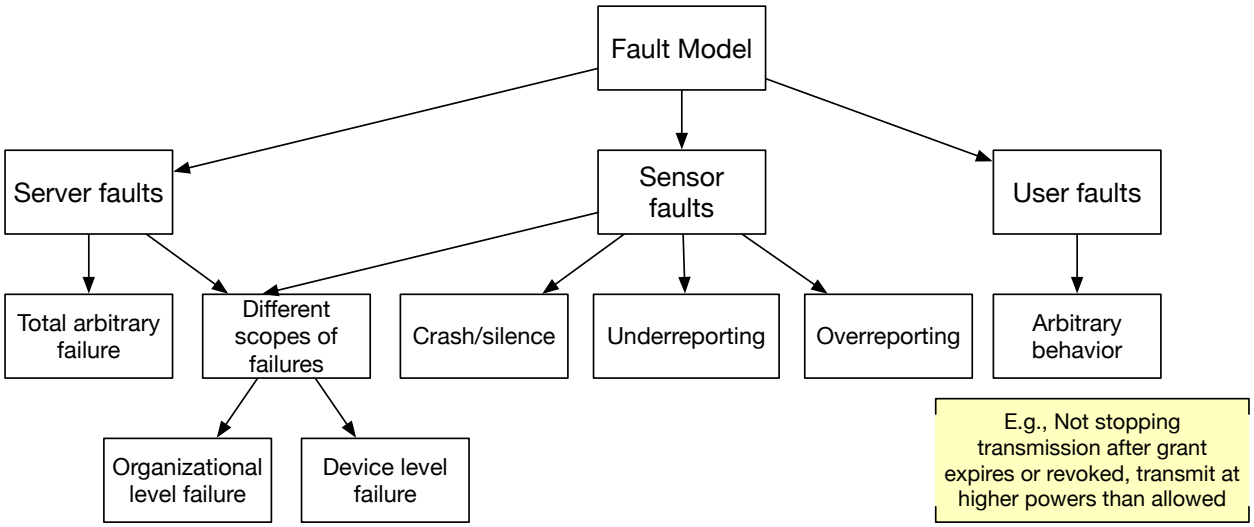


Figure 3: Overview of Faults in Spectrum Sharing System

than servers managing a geographical region, collaboratively determine access to spectrum.

3.2.3 Sensor Model

Environmental sensors detect transmissions from Type 1 users. Upon detecting a Type 1 transmission, a sensor sends a detection report with the transmission characteristics to the servers. As discussed in Section 3.1, sensors may be deployed by the same organizations deploying servers. In terms of our sensor communication model, we can choose to have sensors send their reports to all servers, or to only a subset of servers (e.g. those deployed by the same organization, or a set of collaborating organizations). To tolerate malicious behavior, we should be able to authenticate sensor reports, so that an attacker cannot perform a denial of service attack by simply generating many false detection reports. Similar to the communication model choices, we may consider sensors with global identities (such that all servers can authenticate reports generated by any sensor), or organization-level identities such that servers can only authenticate reports from the sensors assigned to them. In the case of organization-level identities, servers are then responsible for introducing reports to the BFT SMR or interactive consistency protocol on behalf of their sensors and asserting the validity of these reports to the other servers. This has implications for the threat model, since in the organization-level identity case, a compromised server *or* a compromised sensor can introduce false detection reports, while in the global identity case only a compromised sensor can introduce false reports.

3.3 Fault/Threat Models

In Section 3.2, we mainly describe the system model in terms of the standard BFT fault model where up to f servers may be compromised (Byzantine). But, in the spectrum sharing context, this threat model may require extensions to cope with Byzantine *clients* and *organization-level misbehavior*.

In the BFT SMR terminology, users who submit requests and sensors that submit reports are the *clients* of the replicated database servers. One important design consideration is whether/how a system can cope with users or sensors that exhibit malicious behavior. Typical BFT SMR protocols tolerate any number of Byzantine clients in the sense that Byzantine clients cannot cause *disagreement* among servers: all correct servers will execute all operations in the same order. However, they do not address the situation where the content of those operations may be incorrect or malicious [42]. In the spectrum sharing context, faulty or malicious sensors may issue incorrect reports by *over-reporting* spectrum usage (i.e. reported that they have detected a transmission that does not exist). They may also selectively *under-report*, failing to issue reports for some detected transmissions, or remain completely *silent*, failing to issue any reports (in practice, the silent fault type can correspond to a sensor that crashes or experiences a power or hardware failure). Users may also deviate their expected behavior. For example, a user may not stop transmitting after their reservation window ends or after receiving an evacuate command. Users may also transmit with different characteristics than what was requested and approved, e.g., transmitting at a higher power than approved.

Similar to Byzantine server behavior, Byzantine sensor behavior can be tolerated using redundancy. For example, we can design a system to tolerate up to g Byzantine sensors (using g to distinguish it from the number of Byzantine *servers* f). To mask the effects of *over-reporting*, we can require a detection to be reported by at least $g + 1$ sensors (to ensure at least one report is from a correct sensor). But, this has implications for the required density of sensor deployment. To guarantee that $g + 1$ correct sensors will issue a detection report for a given Type 1 transmission, there must be $2g + 1$ sensors in range of that transmission (such that even if g Byzantine sensors remain *silent* or choose to *under-report* and omit the detection report, $g + 1$ correct sensors will detect the transmission and issue reports). In Section 4, we provide a case study analyzing the effects of different sensor deployment and fault models on the number of sensors required to provide adequate detection of Type 1 transmissions.

User misbehavior is more challenging to address, since standard masking-via-redundancy techniques do not apply. Instead, it will likely require augmenting the system with additional mechanisms to detect misbehavior and penalize or disincentivize users. One approach worth considering is using sensors to detect user misbehavior. This approach however will need additional thought especially if users are affiliated with the organizations deploying the sensors. Further, as mentioned in Section 3.1, practical spectrum sharing systems may encounter organization-level Byzantine behavior, including organization-level compromises or insider threats and accidental or malicious misconfigurations. Thus, we may want to extend our threat models to include such organization-level faults. Tolerating such threats requires consideration in the system setup. For example, one approach to tolerate an organization-level fault in a BFT system that tolerates up to f faulty servers is to ensure that no single organization is responsible for more than f servers.

3.4 Performance Metrics

To compare different system designs based on the models above, we must specify the relevant performance metrics. Figure 4 shows our categorization of performance metrics. We consider three classes of metrics: conflicts, latency, and spectrum utilization. The overall goal of the spectrum sharing system is to protect Type 1 users from interference, while maximizing the ability to

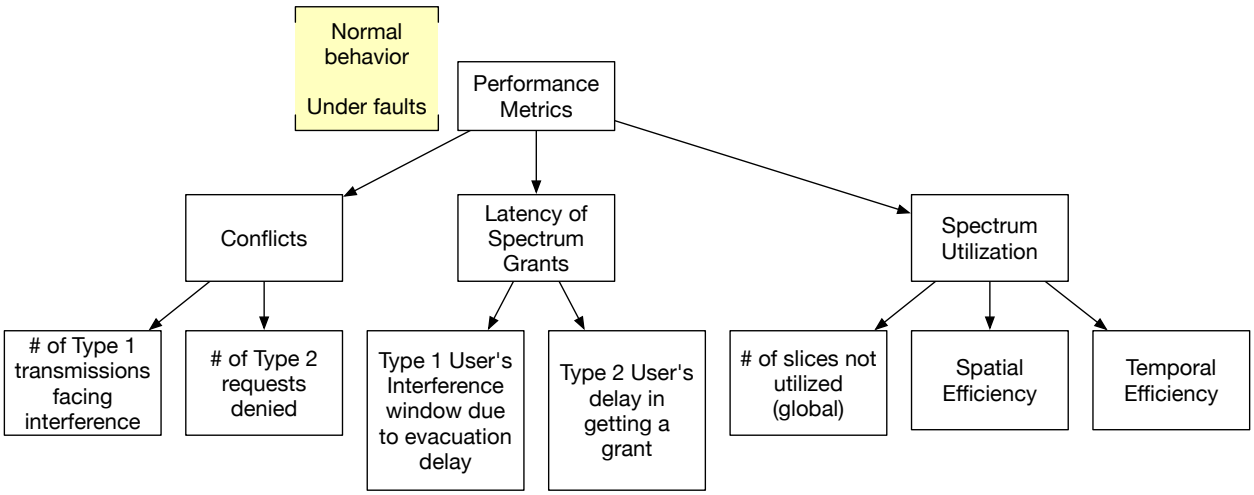


Figure 4: Categorization of Performance Metrics

grant Type 2 user requests, allowing more efficient spectrum usage. In this context, measuring *conflicts* is important to quantify the system's utility to users. Concretely, we propose to measure the number of Type 1 transmissions that experience interference to quantify the effectiveness in protecting Type 1 transmissions. While the goal is to avoid interference, interference may occur if there is an *undetected conflict* between transmissions (e.g. because there are not a sufficient number of correct sensors in range of the Type 1 user's transmission). To measure utility to Type 2 users, we propose to measure the number of Type 2 requests denied. Requests are denied when there is a *detected* conflict between the request and another requested or ongoing transmission.

To provide a more detailed view of utility for both Type 1 and Type 2 users, the *latency* to process reports and requests is another important metric. When a Type 1 user starts transmitting, they may experience temporary interference from ongoing Type 2 transmissions, until the relevant sensors can detect and report the transmission, and the servers can process the report and issue any necessary evacuation commands. Thus, the latency to process *sensor reports* represents the potential *interference window* for detected Type 1 transmissions. For Type 2 users, the latency to process their *requests* represents their *delay in getting a spectrum grant*: to provide a useful service to Type 2 users, we aim to minimize this delay and allow them to transmit as soon as possible.

The conflict and latency metrics capture a user-centric view of the system's effectiveness. In addition, we consider the overall *spectrum utilization* as a metric of the system-level benefits. For this, we propose to measure the global number of slices not utilized, as well as the spatial and temporal efficiency. This captures how effective the system is in enabling increased use of scarce spectrum resources. If spectrum slices are sparsely used in space or time, or see congestion in space or time, it may be possible to devise strategies either to incentivize use or reduce congestion.

Finally, to characterize the *fault tolerance* of a system, we need to measure all of the above aspects both in the **normal fault-free case**, and in the presence of **Byzantine faults**. The difference between these two cases captures the fault tolerance: we aim to minimize the degradation from the fault-free case.

4 Case Study: Density of Environmental Sensors

In this section, we show results of a simulation that examines the *number of environmental sensors* needed for a given level of performance (described below) under a few examples of the different organizational and trust models for sensor deployment discussed above. In particular, we contrast *Global* models where all sensors can communicate with all servers with *Private* models where each server is assigned a subset of the M sensors and only communicates with its own sensors. We contrast *Fully Trusted* models where all sensors are assumed to be correct with *Untrusted* models that assume some sensors may be Byzantine. Note that the Global model for sensor deployment is separate from the Global Agreement model for server coordination described above. Any of the three server coordination models (Global Agreement, Regional Agreement, Dynamic Agreement) can be used with any combination of the sensor deployment models (Global/Private and Fully Trusted/Untrusted). For simplicity, we assume the Global Agreement model for server coordination in our simulation. The simulation and analysis illustrate only a small window of the issues discussed in the paper.

4.1 Sensor Deployment Models

As discussed above, we assume we have N_1 Type 1 and N_2 Type 2 transmitters operating over S slices of a band B. To protect Type 1 users from Type 2 interference, M sensors are deployed uniformly over the geographic region where spectrum sharing takes place. In our simulation, we assume a square region where each side is of length D. We assume sensors have wired connections to the servers they communicate with (so sensor-server communication does not require spectrum assignment or cause interference).

We assume the interaction between sensors and servers works as follows: Each time a sensor detects a Type 1 transmission, it sends a report to its assigned server(s) (which may be all servers in the Global model, or a single server in the Private model). Each server collects sensor reports and applies a local decision procedure to determine when a Type 1 transmission is detected based on the reports (in the Fully Trusted model, the decision is made upon receiving a single report; in the Untrusted model, it requires a sufficient number of reports indicating the same transmission). Upon deciding that a Type 1 transmission is locally detected, the server introduces a detection report for that transmission into the BFT SMR protocol so that it will be ordered and processed by all correct servers in the same way (generating an identical set of evacuate commands, and updating their state at the same logical point in time, such that their future accept/reject decisions on client requests will be identical). In order to tolerate f Byzantine servers, a correct server only acts on a detection report once $f + 1$ matching reports from different servers have been introduced and ordered in the BFT SMR protocol. In this context, *the way sensors are assigned to servers and the level of trust in those sensors create different trade-offs between sensor deployment cost and detection accuracy* for active Type-1 users. We consider four such models and specify the condition required to guarantee that a given Type 1 user's transmission is detected in each:

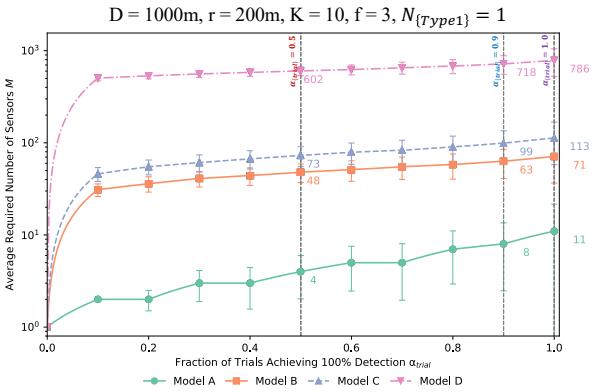
- **Model A (Global-Fully Trusted/Baseline)**: All M sensors report to every one of the K servers, and every server fully trusts all sensor reports. Hence, for each active Type-1 user, it suffices that at least one sensor detects it—once any sensor "sees" the user, all K servers

immediately locally learn of that detection and all correct servers will introduce it to the BFT SMR protocol. Thus, at least $f + 1$ detection reports will be introduced and ordered by the BFT SMR protocol, and the report will be processed at all correct servers.
Tolerated fault model: f Byzantine servers; no Byzantine sensors.
Detection condition: At least one sensor is in range of the Type 1 transmission.
We note that this model also serves as a baseline for the minimum viable number of sensors to cover a given area for *any* system, regardless of fault tolerance considerations, since it only requires one sensor to detect each transmission.

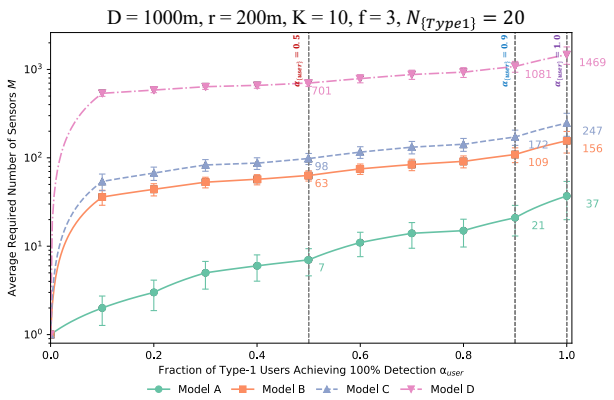
- **Model B (Global-Untrusted)**: As in Model A, all M sensors report to all K servers, but up to g of the sensors may be Byzantine. Each server, therefore, applies a local threshold $T_{sensor} = g + 1$, declaring an active Type 1 user "detected" only if at least $g + 1$ sensors have reported that user. In this setting, to guarantee a user's transmission is detected, it must be in range of at least $2g + 1$ sensors. Since at most g sensors can remain silent or maliciously fail to report, this guarantees that at least $g + 1$ correct sensors remain and will report the transmission. As in Model A, once this detection condition is met, all K servers locally detect the transmission and will introduce it to the BFT SMR protocol, resulting in the report being processed at all correct servers.
  Tolerated fault model: f Byzantine servers; g Byzantine sensors.
  Detection condition: At least $2g + 1$ sensors are in range of the transmission.

- **Model C (Private-Fully Trusted)**: In this model, the M sensors are randomly partitioned into K disjoint cohorts m, and each m is assigned exclusively to one server. Servers fully trust their own sensors (no Byzantine sensor faults). So, locally, each server considers an active Type 1 user detected as soon as any one of its sensors reports detection. To ensure the detection report is processed at all correct servers, at least $2f + 1$ servers must have sensors that report the transmission. This guarantees that at least $f + 1$ *correct* servers locally detect the transmission and introduce it to the BFT SMR protocol.
  Tolerated fault model: f Byzantine servers; no Byzantine sensors.
  Detection condition: At least one sensor *in each of at least* $2f + 1$ *of the K cohorts* is in range of the transmission.

- **Model D (Private Untrusted)**: As in Model C, each server has its own sensor cohort m. However, up to g sensors in each m may be Byzantine. Locally, each server applies a threshold $T_{sensor} = g + 1$, introducing a detection report of an active Type 1 user to the BFT SMR protocol only if at least $g + 1$ of its sensors report the user's transmission. As in Model B, guaranteeing that $g + 1$ sensors report requires that $2g + 1$ sensors are in range of the transmission (so that $g + 1$ *correct* sensors are guaranteed to detect and report it). Globally, to tolerate up to f Byzantine servers, we then require at least $T_{server} = f + 1$ servers to introduce the detection report to the BFT SMR protocol. As in Model C, this requires $2f + 1$ servers to be able to locally detect the transmission to guarantee that $f + 1$ *correct* servers detect and introduce it.
  Tolerated fault model: f Byzantine servers; g Byzantine sensors per cohort.
  Detection condition: At least $2g + 1$ sensors *in each of at least* $2f + 1$ *of the K cohorts* are in range of the transmission.



The number of sensors needed for a given percentage of Monte Carlo trials (out of 5000) to achieve 100% detection of *all* Type-1 users



The number of sensors needed for detecting a given percentage of Type-1 users in *all* Monte Carlo trials

Figure 5: Simulation of the number of sensors needed to maintain BFT (with a detection rate of Type 1 users) with various BFT models.

4.2 Simulation

To illustrate the practical impact of our four sensor deployment models, we fix a square sensing region of side length $D = 1000$ m, deploy $K = 10$ servers, and assume each sensor has a line-of-sight detection radius $r = 200$ m. In all models, up to $f = 3$ of the servers may be Byzantine, and in the Untrusted models, up to $g = 3$ of the sensors may be Byzantine. We consider a setting *with only a single* Type 1 user ($N_1 = 1$) to assess the minimum number of sensors needed to cover the target area (this also maps to practical current scenarios with a single incumbent user), as well as a more general setting with $N_1 = 20$ Type 1 users to assess the fraction of N_1 users that can be detected with a certain number of sensors.

We then perform 5 000 independent Monte Carlo trials. In each trial we:

1. Place N_1 Type-1 users uniformly at random in the $D \times D$ square.

2. Deploy sensors one by one at random locations; after each placement, mark any user within range as "detected."

3. Stop deploying when the model-specific detection criterion is met: **Model A:** Each Type 1 user has been detected by at least one sensor. **Model B:** each user has accumulated at least $2g + 1$ sensor detections. **Model C:** sensors are randomly and evenly split into K disjoint cohorts (one per server); each server declares detection as soon as any sensor in its cohort sees a user, and we require every user to be detected by at least $2f + 1$ different servers. **Model D:** cohorts remain private but up to g sensors per cohort may lie; each server locally detects a user once $2g + 1$ of its own sensors detect that user, and we require every user to be locally detected by at least $2f + 1$ servers.

4. Record per trial: (a) The number of sensors required to detect *all* Type-1 users. (b) The number of sensors required to detect at least a target fraction a_{user} of Type-1 users.

Figure 5 shows the results obtained from this simulation for (1) the minimum sensor budget such that a fraction a_{trial} of trials achieve full detection of all users and (2) average sensor count needed to detect a fraction a_{user} of users in each trial. As an example, in 90% of the simulations, on average 8 sensors were sufficient detect a Type 1 user placed randomly in the square region for Model A, while Model B required an average of 63 sensors for the same scenario.

Specifically, to calculate the sensor budget required such that a fraction a_{trial} of trials achieve full detection of all users, we run the procedure above for 5000 trials, recording the number of sensors required to achieve full detection in each trial. We then sort the results by the number of sensors and report the number of sensors needed such that the target fraction of trials required that specific number or fewer sensors for detection. To calculate the average sensor count needed to detect a fraction a_{user} of users, we simply run the 5000 trials, recording the number of sensors needed to detect the given fraction of users in each trial, and average the results across all trials.

5 Discussion and Conclusion

We explored how BFT systems can support decentralized coordination in dynamic spectrum sharing regimes. Our approach brings together technical models from distributed computing with institutional perspectives on property rights, offering a hybrid framework for resilient, polycentric spectrum governance. We have shown that BFT protocols can ensure consistency and reliability in spectrum grant decisions even in adversarial environments, making them well-suited for contexts where multiple, potentially untrusted parties must share a contested resource. By framing spectrum access as a distributed decision problem—and by allowing for dynamic, revocable grants—BFT systems help shift the paradigm from static licensing to responsive coordination, echoing Ostromian principles of adaptive, rule-based management of shared resources.

Comparatively, BFT-based systems address many of the shortcomings of both centralized and unstructured unlicensed regimes. Unlike centralized coordination schemes such as the SAS in CBRS, BFT protocols avoid single points of failure and can scale across competing organizations while maintaining consistency and fault tolerance. Unlike traditional unlicensed models that rely on device-level sensing (e.g., DFS), BFT systems offer a more robust mechanism for adjudicating conflicting claims in real time using structured consensus. Moreover, our simulations show that BFT designs can accommodate various organizational models—ranging from fully trusted to adversarial sensor deployments—while maintaining performance guarantees under minimal assumptions.

That said, there are limitations with BFT approaches to spectrum sharing. First, BFT protocols typically involve significant communication overhead and can suffer from latency as the number of participants or geographic span increases. This challenge is particularly acute in the Global Agreement model. Second, the effectiveness of BFT systems depends on accurate sensing and honest reporting from environmental sensors. While our models account for sensor-level faults, deploying sufficient numbers of sensors to guarantee timely detection—especially under private or adversarial trust models—can be costly and logistically demanding. Third, while BFT ensures agreement among servers, it does not itself resolve broader institutional questions about rule-making, rights enforcement, or conflict resolution. These governance functions remain essential, particularly in environments with overlapping jurisdictions or contested legitimacy.

Despite these challenges, the BFT approach is promising for several reasons. First, it aligns with the trend toward decentralized, service-based models of spectrum access, where dynamic grants substitute for long-term licenses. Second, BFT allows communities or organizations to implement their own internal rules while interoperating through shared consensus protocols providing a foundation for both endogenous and exogenous rule integration (polycentricity). Finally, BFT protocols are compatible with modular, open architectures: they can be embedded in systems governed by spectrum-as-a-service platforms, used alongside cryptographic auditing tools, or integrated with broader institutional safeguards (e.g., FCC-enforced priority rights or data registries). In this sense, BFT offers not just a technical fix, but an enabling infrastructure for more adaptive and trustworthy spectrum governance.

Ultimately, while BFT offers a technically rigorous framework for decentralized coordination, its promise can only be realized within a supportive institutional environment. The successful operation of BFT protocols presupposes not only technical inputs such as accurate sensing, authenticated communication, and consensus algorithms, but also institutional features such as rule clarity, enforcement capacity, and autonomy to experiment with new governance models. Spectrum sharing, like other forms of commons governance, is not merely a technical engineering problem but an institutional design challenge. BFT illustrates that even the most advanced coordination technologies must be embedded in a system that allows for local experimentation, polycentric rule-making, and dynamic adaptation to evolving conditions. In this way, BFT is not just a mechanism of agreement, it is a test case for the broader insight that innovation in spectrum governance depends as much on institutional flexibility as on technical ingenuity.

Acknowledgments

This work was funded in part by a subaward (seed grant) from SpectrumX: An NSF Spectrum Innovation Center (NSF Award No. 2132700). The authors would like to thank Randall Berry, Darrah Blackwater, Michael Honig, Monisha Ghosh, Dongning Guo, Thomas Hazlett, Ali Palida, and Martin Weiss for the many discussions on spectrum sharing, policy, native American rights, and other aspects of rights, economics, and policy of spectrum sharing. This work would not have been possible without the conversations in the corresponding research community in SpectrumX. Prashant Krishnamurthy would like to express his thanks to David Tipper for stimulating conversations on spectrum usage, networks, and policy. Ilia Murtazashvili acknowledges support from the Center for Governance and Markets at the University of Pittsburgh.

References

[1] A. Abouzeid *et al.*, "National spectrum strategy research and development plan," *National Science and Technology Council*, October 2024.

[2] M. Castro and B. Liskov, "Practical byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems*, vol. 20, no. 4, pp. 398–461, 2002.

[3] T. W. Hazlett, "Spectrum tragedies," *Yale Journal on Regulation*, vol. 22, pp. 242–274, 2005.

[4] T. W. Hazlett, "Optimal abolition of fcc spectrum allocation," *Journal of Economic Perspectives*, vol. 22, no. 1, pp. 103–128, 2008.

[5] T. W. Hazlett, D. Porter, and V. Smith, "Radio spectrum and the disruptive clarity of ronald coase," *The Journal of Law and Economics*, vol. 54, no. S4, pp. S125–S165, 2011.

[6] E. Schlager and E. Ostrom, "Property-rights regimes and natural resources: A conceptual analysis," *Land Economics*, pp. 249–262, 1992.

[7] T. W. Hazlett, "Tragedy tv: Rights fragmentation and the junk band problem," *Arizona Law Review*, vol. 53, pp. 83–130, 2011.

[8] R. H. Coase, "The problem of social cost," *Journal of Law and Economics*, vol. 3, pp. 1–44, 1960.

[9] E. Ostrom, *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, 1990.

[10] P. Bustamante, M. Gomez, W. Lehr, I. Murtazashvili, A. Palida, and M. B. Weiss, "Examining the us amateur-radio community through a polycentricity lens," *Telecommunications Policy*, vol. 47, no. 10, p. 102667, 2023.

[11] E. Ostrom, *Understanding Institutional Diversity*. Princeton: Princeton University Press, 2005.

[12] T. W. Hazlett, *The Political Spectrum: The Tumultuous Liberation of Wireless Technology, from Herbert Hoover to the Smartphone*. New Haven: Yale University Press, 2017.

[13] P. Bustamante, W. Lehr, I. Murtazashvili, A. Palida, M. B. Weiss, and M. Gomez, "Polycentric governance in the amateur radio community: Unassigned spectrum and promoting open innovation," in *Telecommunications Policy Research Conference (TPRC)*, (Washington, D.C.), 2022.

[14] Ofcom, "Enabling short notice short duration licences in 2.3 GHz," *Consultation*, July 1 2025.

[15] OnGo Alliance, "Collaborative GAA coexistence technical specification," *ONGO TS-2003*, December 5 2023.

[16] L. Herzel, "My 1951 color television article," *The Journal of Law and Economics*, vol. 41, no. S2, pp. 523–528, 1998.

[17] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, 1982.

[18] M. Singhal and N. G. Shivaratri, *Advanced concepts in operating systems*. McGraw-Hill, Inc., 1994.

[19] M. J. Fischer, "The consensus problem in unreliable distributed systems (a brief survey)," in *Foundations of Computation Theory* (M. Karpinski, ed.), (Berlin, Heidelberg), pp. 127–140, Springer Berlin Heidelberg, 1983.

[20] M. van Steen and A. S. Tanenbaum, *Distributed Systems, 4th ed.* distributed-systems.net, 2023.

[21] M. Kleppmann, "Distributed systems (lecture notes)." `https://www.cl.cam.ac.uk/teaching/2122/ConcDisSys/dist-sys-notes.pdf`, 2021.

[22] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *J. ACM*, vol. 32, p. 374–382, Apr. 1985.

[23] P. Civit, S. Gilbert, R. Guerraoui, J. Komatovic, and M. Vidigueira, "On the validity of consensus," in *Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing*, PODC '23, (New York, NY, USA), p. 332–343, Association for Computing Machinery, 2023.

[24] F. B. Schneider, "The state machine approach: A tutorial," in *Fault-Tolerant Distributed Computing* (B. Simons and A. Spector, eds.), (New York, NY), pp. 18–41, Springer New York, 1990.

[25] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: Speculative byzantine fault tolerance," *ACM Trans. Comput. Syst.*, vol. 27, Jan. 2010.

[26] C. Berger, L. Rodrigues, H. P. Reiser, V. Cogo, and A. Bessani, "Chasing lightspeed consensus: Fast wide-area byzantine replication with mercury," in *Proceedings of the 25th International Middleware Conference*, Middleware '24, (New York, NY, USA), p. 158–171, Association for Computing Machinery, 2024.

[27] Y. Amir, B. Coan, J. Kirsch, and J. Lane, "Prime: Byzantine replication under attack," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 4, pp. 564–577, 2011.

[28] A. Clement, E. Wong, L. Alvisi, M. Dahlin, M. Marchetti, *et al.*, "Making byzantine fault tolerant systems tolerate byzantine faults," in *Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, pp. 153–168, The USENIX Association, 2009.

[29] Z. Milosevic, M. Biely, and A. Schiper, "Bounded delay in byzantine-tolerant state machine replication," in *2013 IEEE 32nd International Symposium on Reliable Distributed Systems*, pp. 61–70, 2013.

[30] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "Hotstuff: Bft consensus with linearity and responsiveness," in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, PODC '19, (New York, NY, USA), p. 347–356, Association for Computing Machinery, 2019.

[31] G. Golan Gueta, I. Abraham, S. Grossman, D. Malkhi, B. Pinkas, M. Reiter, D.-A. Seredinschi, O. Tamir, and A. Tomescu, "Sbft: A scalable and decentralized trust infrastructure," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 568–580, 2019.

[32] M. B. H. Weiss, K. Werbach, D. C. Sicker, and C. E. C. Bastidas, "On the application of blockchains to spectrum management," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 2, pp. 193–205, 2019.

[33] L. Perera, P. Ranaweera, S. Kusaladharma, S. Wang, and M. Liyanage, "A survey on blockchain for dynamic spectrum sharing," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 1753–1802, 2024.

[34] A. Bessani, E. Alchieri, J. Sousa, A. Oliveira, and F. Pedone, "From byzantine replication to blockchain: Consensus is only the beginning," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 424–436, 2020.

[35] Y. Xiao, S. Shi, W. Lou, C. Wang, X. Li, N. Zhang, Y. T. Hou, and J. H. Reed, "Bd-sas: Enabling dynamic spectrum sharing in low-trust environment," *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 4, pp. 842–856, 2023.

[36] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "Trustsas: A trustworthy spectrum access system for the 3.5 ghz cbrs band," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 1495–1503, 2019.

[37] L. Zhang, G. Ding, Q. Wu, Y. Zou, Z. Han, and J. Wang, "Byzantine attack and defense in cognitive radio networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1342–1363, 2015.

[38] A. Mollakhani and D. Guo, "Fault-tolerant spectrum usage consensus for low-earth-orbit satellite constellations," 2024.

[39] M. Weiss, P. Krishnamurthy, L. E. Doyle, and K. Pelechrinis, "When is electromagnetic spectrum fungible?," in *2012 IEEE International Symposium on Dynamic Spectrum Access Networks*, pp. 349–357, 2012.

[40] G. Wang, Z. J. Shi, M. Nixon, and S. Han, "Sok: Sharding on blockchain," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, AFT '19, (New York, NY, USA), p. 41–61, Association for Computing Machinery, 2019.

[41] C. Berger, S. Schwarz-Rüsch, A. Vogel, K. Bleeke, L. Jehl, H. P. Reiser, and R. Kapitza, "Sok: Scalability techniques for bft consensus," in *2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 1–18, 2023.

[42] Y. Amir, C. Danilov, J. Lane, M. Miskin-Amir, and C. Nita-Rotaru, "Enhancing distributed systems with mechanisms to cope with malicious clients," tech. rep., Technical Report CNDS-2005-4, the Distributed Systems and Networks Lab, Johns Hopkins University, 2005.