Building Flexible, Low-Cost Wireless Access Networks With Magma

Shaddi Hasan, *Virginia Tech;* Amar Padmanabhan, *Databricks;* Bruce Davie, *Systems Approach;* Jennifer Rexford, *Princeton University;* Ulas Kozat, Hunter Gatewood, Shruti Sanadhya, Nick Yurchenko, Tariq Al-Khasib, Oriol Batalla, Marie Bremner, Andrei Lee, Evgeniy Makeev, Scott Moeller, Alex Rodriguez, Pravin Shelar, Karthik Subraveti, Sudarshan Kandi, Alejandro Xoconostle, and Praveen Kumar Ramakrishnan, *Meta;* Xiaochen Tian, *Indepenent;* Anoop Tomar, *Meta*

https://www.usenix.org/conference/nsdi23/presentation/hasan

This paper is included in the
Proceedings of the 20th USENIX Symposium on
Networked Systems Design and Implementation.

April 17–19, 2023 • Boston, MA, USA

Building Flexible, Low-Cost Wireless Access Networks With Magma

Shaddi Hasan[1], Amar Padmanabhan[2], Bruce Davie[3], Jennifer Rexford[4], Ulas Kozat[5], Hunter Gatewood[5], Shruti Sanadhya[5], Nick Yurchenko[5], Tariq Al-Khasib[5], Oriol Batalla[5], Marie Bremner[5], Andrei Lee[5], Evgeniy Makeev[5], Scott Moeller[5], Alex Rodriguez[5], Pravin Shelar[5], Karthik Subraveti[5], Sudarshan Kandi[5], Alejandro Xoconostle[5], Praveen Kumar Ramakrishnan[5], Xiaochen Tian[6], and Anoop Tomar[5]

[1]Virginia Tech   [2]Databricks   [3]Systems Approach   [4]Princeton   [5]Meta   [6]Independent

Abstract

Billions of people remain without Internet access due to availability or affordability of service. In this paper, we present Magma, an open and flexible system for building low-cost wireless access networks. Magma aims to connect users where operator economics are difficult due to issues such as low population density or income levels, while preserving features expected in cellular networks such as authentication and billing policies. To achieve this, and in contrast to traditional cellular networks, Magma adopts an approach that extensively leverages Internet design patterns, terminating access network-specific protocols at the edge and abstracting the access network from the core architecture. This decision allows Magma to refactor the wireless core using SDN (software-defined networking) principles and leverage other techniques from modern distributed systems. In doing so, Magma lowers cost and operational complexity for network operators while achieving resilience, scalability, and rich policy support.

1 Introduction

Good Internet connectivity has become a basic necessity for people and enterprises all over the world. Yet, more than one-third of the global population does not have access to the Internet [54], and many other users do not have the high-speed connectivity needed for many important applications. The problem is primarily a matter of economics: commercial network operators claim that today's Internet has reached the user footprint that seems commercially viable to serve [29]. To reach the next billion users, we must reduce the *cost* of providing Internet access or enable actors beyond traditional, large-scale commercial operators to build sustainable, scalable network infrastructure. We need effective ways to reduce both capital and operational costs, through less expensive equipment and software, less reliance on highly skilled network administrators, and increased utilization of existing local capabilities. At the same time, providers need ways to manage their limited network resources effectively to enable sustainable network operation. Cellular networks typically achieve

these goals with per-user policies, which may include per-user data caps, rate limits, or usage-based charging.

Unfortunately, conventional wireless solutions are not well suited to many scenarios affecting under-served users. WiFi access points operating on unlicensed spectrum cannot generally provide efficient coverage to large geographic regions (e.g., sparsely populated rural areas) due to the propagation characteristics of the radios. Plus, WiFi networks typically do not offer fine-grained policies to manage resources. In contrast, cellular base stations offer wider coverage, support more users, and connect to core networks that support more flexible policies. However, today's cellular access networks rely on expensive equipment, complex protocols, and a highly skilled workforce, limiting their ability to cost-effectively connect the next billion. While cellular networks scale *up* to large user populations, they do not scale *down* well. That is, a small cellular deployment is typically quite expensive. Magma aims to bridge the gap between these two classes of solution: cellular networks with rich policies, large user populations, and long distances, and the simpler but less scalable WiFi networks.

More fundamentally, we observe that choosing to use a cellular radio access network (RAN) today *forces* a network operator to make a series of decisions that deeply impact their network operations that are not inherently related to their choice of access network technology. This choice binds a network operator to: (i) a specific network architecture—namely the 3GPP-defined arrangement of interfaces for network management and on-path devices for policy enforcement, (ii) an ecosystem of vendors that has largely evolved to meet the needs of massive-scale telecom operators, and (iii) a particular set of radio frequencies and associated regulatory requirements. The Magma project aims to change all this, by creating an open-source, carrier-grade wireless networking platform that supports a wide range of deployment scenarios. Magma deployments can leverage whatever radio access technology is readily available and most appropriate for their density of subscribers or deployment scenario. Magma achieves this goal through *access gateways* that terminate the radio-specific protocols as close to the radios as possible. As a result, Magma

allows carriers to augment an existing cellular deployment with WiFi hotspots in popular locations (e.g., athletic venues), or use LTE base stations to serve homes in rural areas, using a single core network and management platform.

Ideally, new deployments could start small and grow over time. Magma achieves a "scale as you go" design through horizontal scaling of software components that run on commodity hardware, as is common in cloud-computing environments. Magma also leverages open-source software components (e.g., Open vSwitch, gRPC, Kubernetes, Prometheus) commonly used in cloud settings. Magma simplifies network management by adopting software-defined networking concepts, so that a central point of control can be used to set network policies, manage subscribers, etc. Magma adopts a hierarchical control plane to improve scalability. Magma supports only the essential features for efficient Internet access (e.g., authentication, accounting, and per-user policies), and forgoes some complex features. For example, while Magma supports both mobility (within the area served by an access gateway) and roaming, it does not yet support seamless user mobility *between* access gateways; this is because mobility has not been a requirement for the use cases that commercial deployments of Magma support (e.g., home broadband or backhaul to WiFi hotspots). As other work has observed [38], modern end-host protocols and applications can perform well without in-network mobility support.

In this experiences paper, we present the lessons learned in designing and deploying Magma. We discuss how the goals of supporting heterogeneous radio and backhaul technologies and flexible policies, all at low cost, lead to a novel software architecture. Magma is used in real-world deployments that vary significantly in geographic scope, number of users, technology choices, and the business models that make them financially sustainable. In Section 2 we motivate Magma's central tenet that the radio access technology should not dictate the network architecture. Then, Section 3 discusses how the design of the access gateways enables Magma to support diverse technologies, a scalable control plane, fault tolerance, and more. Next, Section 4 presents an experimental evaluation that demonstrates that Magma design and implementation achieves good performance and scalability along with a discussion of two production access networks. We have seen cost savings in one deployment of 43% compared to traditional approaches due to lower operational, hardware, and software costs. Our deployment experience also illustrates how Magma scales both up and down, with one deployment supporting more than 800 eNodeBs (base stations) in 45 US states at the time of writing. Section 5 presents related work. The paper concludes in Section 6 with a discussion of ongoing work on Magma and future challenges.

Ethics. This paper raises no ethical concerns. For the deployments discussed in Section 4, we only consider operational data and did not have access to any user data or traffic.

2 The Radio Access Technology Should Not Drive the Network Architecture

Traditionally, the choice of radio access technology dictates a raft of other decisions about the network architecture. In contrast, Magma starts with the premise that each radio access technology has a role to play in reaching diverse user communities and that network operators should be able to use the radio and backhaul technologies most suited for a deployment scenario. In short, wireless network architectures should, like the Internet itself, abstract away the link layer.

2.1 WiFi vs. Cellular Access Networks

The two main classes of radio access technologies emerged as extensions to existing wireline networks with different design philosophies. WiFi extended IP networks, whereas modern cellular data networks began as extensions to voice telephony networks. Many of the differences between these two classes of access networks follow directly from this early distinction.

WiFi: WiFi allows inexperienced users to run simple low-cost local-area networks on their own. These networks use unlicensed radio spectrum (typically at 2.4 GHz and 5 GHz) that do not require WiFi network operators to get advance regulatory approval. At the same time, anyone can access the same spectrum, subject to limits on transmission power. As a result, WiFi networks share their bands with devices including baby monitors, cordless phones, and smart power meters, so the WiFi MAC layer must assume that a WiFi access point (AP) operates in the presence of physical-layer interference. Combined with power restrictions that limit transmit distance, WiFi is most suitable for dense coverage in small areas. WiFi service is best-effort, consistent with the Internet design philosophy—and realistic given the likelihood of interference. Enterprise WiFi deployments, such as those on college campuses and in corporate office buildings, perform more centralized management of interference across multiple overlapping access points. Still, the risk of interference means that the service remains best-effort.

Cellular: Cellular access networks allow telecommunication providers to offer wireless service to their subscribers, typically using licensed spectrum that is owned or leased by the carrier for long periods of time at high cost. Since the radio has exclusive access to spectrum over a geographic region, cellular waveforms are designed for wide-area coverage and high spectral efficiency, with deployments by well-resourced actors that can acquire land, build and connect towers, and hire skilled staff.

Regardless of access technology, any network of significant scale requires substantial investment in equipment, staffing, and, in the case of cellular networks, regulatory licenses. Thus, beyond very small networks, operators implement policies to manage limited spectrum, ranging from access control; charging for service based on time, usage, or more sophisticated
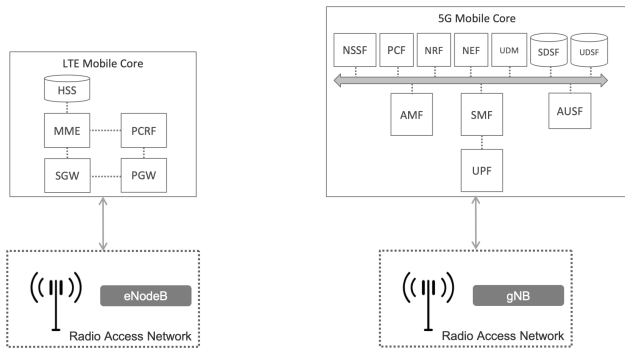
Figure 1: Differences between the LTE and 5G architectures. See Appendix for explanation of acronyms.
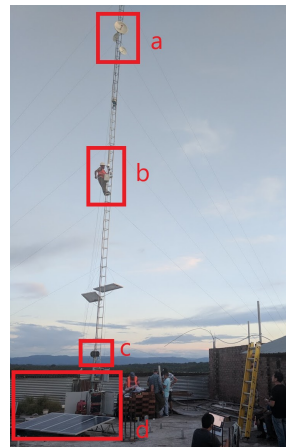

Figure 2: An early Magma deployment with a small rural ISP in Peru (their first cellular site). Components (top to bottom) include (a) point-to-point wireless backhaul, (b) LTE radio and antenna, (c) ruggedized embedded PC serving as Magma AGW, and (d) solar power and battery backup for site.

techniques that incorporate community values [36]; usage caps; and throttling. The policy specification for LTE, for example, runs to almost 300 pages [14]. A simple example policy would be: "rate limit customer C to X Mbps until they have sent Y GB in interval t_1, then limit to Z Mbps for interval t_2." Supporting flexible policies can help carriers reach under-served users in a financially sustainable manner: even networks operating for social reasons still incur costs and must efficiently manage limited resources.

These capabilities are implemented by a sophisticated packet core network that connects multiple base stations to the Internet. In contrast to how the Internet architecture changes incrementally, each generation of cellular network has been an opportunity to rethink everything from authentication to the modularity of the control and data planes.

As such, different generations of the 3GPP standards [15] have different packet core architectures. UTMS ("3G") differs from LTE ("4G"), which differs from 5G, and all of the generations differ from enterprise WiFi. The differences between LTE and 5G are illustrated in Figure 1, adapted from [46]. The different radio technologies require differences in the base stations (eNodeB versus gNB) but note also the change in modularity of the mobile core. WiFi would be different again, and less standardized, with functions such as Authorization, Authentication, and Accounting (AAA) corresponding roughly to Mobility Management Entity (MME) and Home Subscriber Server (HSS) components in LTE.

Today, the boundaries between cellular and WiFi are increasingly blurry, with operators deploying each technology in scenarios more classically served by the other. In recent years, large WiFi deployments have adopted more sophisticated methods for user authentication, power control, seamless mobility, and more [23,24,60], with efforts like Eduroam [58] and OpenRoaming [57] bringing cellular-like wide area roaming to users of WiFi access networks. Similarly, some cellular access networks now use "lightly licensed" spectrum, such as Citizen's Band Radio Service (CBRS) [17] that supports dynamic allocation of radio spectrum to give radios exclusive access to some portion of the spectrum (on the timescales of tens of minutes). Enterprises are deploying *private* cellular access networks for a range of use cases—such as industrial automation, medical applications, and Internet access at hotels and sporting events—that need better radio efficiency, authentication, and performance than WiFi traditionally offers.

2.2 Lowering the Barriers

Magma aims to lower the barrier to connecting under-served populations via wireless networks. We argue that operators should be able to choose the appropriate access technology for any deployment without then being locked into a core architecture that is compatible only with that access type. A single design that supports heterogeneous technologies amortizes the substantial engineering effort for creating software and the costs of training and supporting those who operate the networks. Plus, the design enables a single carrier to use multiple radio technologies (e.g., WiFi in shopping malls and cellular elsewhere) on a single core.

Cellular access has high barriers to *entry*. Network operators deploying cellular access technologies must make large capital investments (CapEx) in infrastructure: whereas a WiFi access point can cost under US$100, even a low-cost cellular deployment would cost at least 1-2 orders of magnitude more. Traditionally, cellular core network equipment is designed for large deployments with hundreds of base stations and does not "scale down" to small initial deployments at reasonable cost; these networks also have high operational costs (OpEx), relying on highly skilled staff to manage the equipment. In addition, remote communities may not have affordable access to the high-quality, low-latency backhaul (e.g., fiber) links cellular networks typically rely upon. Instead, these networks may use satellite or wireless backhaul links with lower reliability and performance.

In contrast, WiFi deployments can start small, but present high barriers to *scale*. WiFi networks do not typically require skilled staff to deploy. Because WiFi is an inherently best-

effort access technology, these networks can leverage any available backhaul, even ad-hoc mesh backhaul using the same physical WiFi radios. Yet WiFi networks do not typically offer scalable ways to implement network policy or (beyond proprietary and vendor-specific solutions) to manage large networks. Thus, it is difficult for a WiFi-oriented operator to offer financially sustainable service over a wide area along the lines of large (usually cellular) operators.

Despite the differences between WiFi and cellular, these barriers are *not* fundamental. The building blocks of network policies are common in each; what is missing is architectural support. Software-defined networking can help address these gaps by enabling network-wide control over a distributed infrastructure, and adopting "scale out" techniques based on commodity components can reduce cost. In short, adopting and extending successful Internet and cloud approaches to scalability and management can make it possible to create a wireless access network that is both flexible and affordable.

3 Magma Architecture

Magma cannot overcome the shortcomings of existing solutions simply by reimplementing a standard, 3GPP-compliant mobile core. Instead, Magma terminates the radio-specific protocols as early as possible, in *access gateways* (AGWs) connected directly to the radio access network, as shown in Figure 3. These access gateways are instrumental in handling a variety of radio technologies in a single design. The Magma architecture goes beyond the traditional RAN/core split of 3GPP to place additional functionality in the access gateway, with a goal of making the packet core more scalable, including scaling down. Notably, Magma adopts the architecture of software-defined networking (SDN) systems, using a hierarchical control-plane design where a local controller in each access gateway interacts with a centralized *orchestrator*. The orchestrator is the central point of control for the system and maintains authoritative state related to system-wide configuration (*config* state). Runtime state, which relates to the activity of user equipment (UEs), is localized to the AGW that serves the appropriate base station for a given UE.
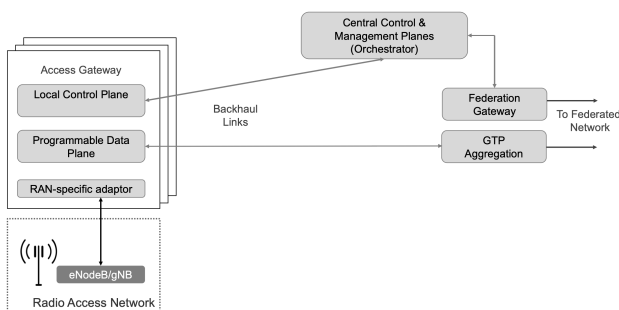

Figure 3: Simplified Magma architecture

Each AGW is a small fault domain, ensuring that the failure or upgrade of any one component affects relatively few users. In this way, Magma's architecture is similar to modern cloud systems designed to run on low-cost hardware that is prone to failure [26]. Magma adopts other ideas from cloud architectures, including the use of gRPC for communication among components, a "desired state" model for state synchronization, and software-based, programmable data plane. While common in cloud computing deployments, these decisions deviate significantly from the way typical 3GPP networks are designed and managed.

3.1 Abstracting the Radio Access Technology

As Figure 1 illustrates, the details of the radio access technology traditionally "leak" into the core network. To counter this, Magma identifies a core set of functions that the AGW must implement for any radio technology (e.g., finding the appropriate policy for a given subscriber) and provides them in an access-technology-independent way. These functions form the heart of an AGW, as illustrated on the right side of Figure 4. Control protocols, which are specific to a given radio technology, are terminated *early* in technology-specific modules close to the radio. These modules, on the left of the figure, communicate with the generic functions (e.g., subscriber management, access control and management) on the right using messages that are RAN-agnostic.

Consider the example of "attaching" a newly active UE. The UE communicates with a nearby base station over a temporary (unauthenticated) radio link. In traditional 4G implementations, the base station forwards the request to the Mobility Management Entity (MME), which initiates an authentication protocol with the UE. The MME consults a subscriber database, authenticates the UE, creates an entry in a session table, and informs the other components of the parameters needed to serve the UE including: (a) assigning an IP address to the UE and setting the appropriate QoS parameters in the data plane; (b) instructing the base station to establish an encrypted channel to the UE; and (c) giving the UE the symmetric key for the encrypted channel. At the end of this sequence of events, the UE has an active *session* established with the mobile core and is able to send and receive data.

These functions are performed in the Magma AGW in a way that abstracts the details of the radio technology, as illustrated on the right-hand side of Figure 4. For example, Magma's subscriber database has the union of all capabilities across the radio access types, even if some fields in a given database row are valid only for some technologies. QoS policies, for example, are less rich in WiFi than in 4G networks, while 4G policies are in turn less rich than those of 5G. Similarly, UE authentication and session establishment are done in a common way by generic functions that cover 4G, 5G, and WiFi procedures. The data plane, which is implemented in different devices across 4G, 5G, and WiFi, is implemented in

Magma	LTE	5G	WiFi
Access Control/Management	MME	AMF	RADIUS AAA
Subscriber Management	HSS	UDM/AUSF	RADIUS AAA
Session/Policy Management	MME/PCRF	SMF/PCF	RADIUS AAA
Data Plane Configuration	SGW/PGW	SMF	WiFi data plane
Data Plane	SGW/PGW	UPF	WiFi data plane
Device Management	per-box configuration		
Telemetry and logging	no equivalent defined		

Table 1: Magma abstractions vs. RAN-specific versions

a common, programmable data plane for Magma.

Table 1 shows how the various components of 4G, 5G, and WiFi are all mapped onto a common set of Magma abstractions. The key observation here is that there are a certain set of functions that need to be performed to authenticate users, establish session state, control the data plane, and so on. Magma does all of these in a generic way that is agnostic to the radio technology in use, thus providing an implementation in which the radio-specific details are abstracted from the core and limited to protocol termination close to the radio itself.

Additionally, Magma adds some generic functions that are not part of the 3GPP standards: device management and telemetry. Coupled with the SDN architecture, this simplifies the management of a large number of devices spread over a wide geographical area. Rather than logging into a specific device to configure it or check its statistics, Magma provides central management and monitoring from the orchestrator, where it can be leveraged by other systems that consume the northbound API. We have found that considering device management and telemetry as first-class responsibilities of Magma significantly reduces the operational complexity and cost of operating access networks (Section 4.3.1).

We do not claim that Magma's decomposition of functionality (Figure 4) is fundamental, but our operational experience shows that it is useful both from an engineering perspective and for a wide range of use cases (as discussed further in Section 4). The modularity between components allows Magma's internal interfaces to evolve independently of the RAN, aligning with an iterative development approach and in stark contrast to rigid 3GPP interface definitions. This has enabled the team to perform major changes to AGW functionality, such as adding new features (e.g., 5G support) or refactoring internal services without exposing these changes southbound toward the RAN or northbound toward the orchestrator API.

All communication between the RAN-specific modules on the left of Figure 4 and the generic functions on the right use gRPC [5], an open-source Remote Procedure Call (RPC) framework, as does all long-distance communication (e.g., from the AGW to the orchestrator). Although this is a typical approach for building modern distributed systems, it differs substantially from the protocols defined for communication among 3GPP components, which leak endpoint (e.g., UE and MME) consistency requirements into a network-level protocol. By running over HTTP, gRPC inherits the resilience to loss and delay of TCP/IP, which is absent from some 3GPP
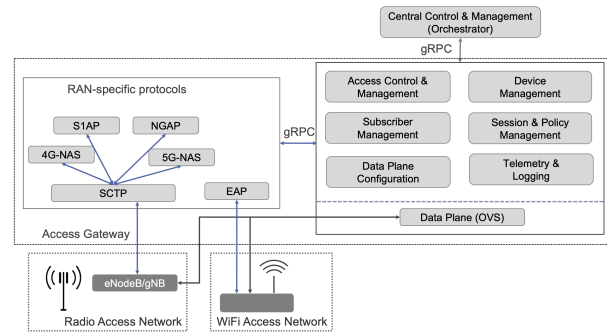

Figure 4: Common functions and RAN-specific protocols in the Magma architecture.

protocols designed for more benign, controlled environments (e.g., leased lines). A concrete example is GTP (GPRS Tunneling Protocol), which is sensitive to loss and latency to the point that it struggles to operate over lower quality or congested backhaul links, such as satellite or shared microwave links. Thus, adopting gRPC allows Magma more latitude to implement alternative consistency models without breaking UE state machines in a wider range of backhaul network conditions. In practice, this tolerance helps mitigate poor error handling on devices: while UEs should reconnect after experiencing a 3GPP protocol-level failure, we find that UEs with low-end baseband processors do not do so reliably. When a UE fails to reconnect, the failure manifests as a confusing lack of coverage to people using these devices, and the failure typically only resolves after power cycling the UE. Since Magma terminates GTP locally in the AGW without traversing the backhaul link, a UE never sees a dropped GTP connection and does not have to handle the error.

While agnostic to the radio technology, Magma necessarily makes practical choices about the order of feature development. Many early Magma deployments used LTE, so we have prioritized support for LTE features, with 5G support coming later. A good example is the support of QoS policies. Simple policies to impose rate limits and usage caps, as outlined in Section 2.2, are supported today for both LTE and 5G. More complex policies could be expressed, particularly in 5G, but full support for richer policies is currently under development.

3.2 Hierarchical SDN Control Plane

Magma adopts software-defined networking (SDN) to reduce operational complexity and minimize reliance on skilled staff. Rather than configuring a distributed collection of devices, providers specify network-wide policies at the orchestrator. The orchestrator provides a central point of control and exposes a northbound API for integration with other systems (e.g., for metrics, alerting, and monitoring). However, running the entire control plane in a central controller would impose limits on the scalability of the system. Hence, practical SDN

systems like Network Virtualization Platform (NVP) [37] and Open Virtual Network (OVN) [42] adopted a hierarchical control plane, and this is the model used by Magma.

In a hierarchical control-plane design, we identify those elements of the control plane that have network-wide scope; these are candidates for the central controller. For example, to add a new subscriber to the network, the long-lived information about the subscriber is network-wide information that is created and maintained by the central controller. Conversely, much of the runtime state associated with a UE can be localized to a single AGW. For example, upon becoming active, a UE is associated with a single AGW. The UE's session state can be created and managed by the local control plane of that AGW. Thus, much of the control plane is able to scale out with increasing numbers of base stations and subscribers, rather than increasing the processing in the central controller.

This division between central and local control planes roughly corresponds to the timescale of changes to the control-plane state. The addition of a new subscriber happens on configuration timescales, and that state is managed centrally. The creation of session state—when a UE becomes active and attaches to an AGW—happens more frequently. This runtime state is handled by the local controller on an AGW.

As with any SDN architecture, we must consider "headless" operation, i.e., the situation where a data plane node is disconnected from the central control plane. In a traditional SDN approach, the goal is to ensure that the data plane continues to operate without the control plane, even as updates to the data plane may be impossible while the control plane is disconnected. With a hierarchical control plane, many local operations are still possible even while the central controller is unreachable. For example, an AGW can still establish a session for a UE that attaches to a base station, because the local control plane has enough information (e.g., cached subscriber profiles) to process the session establishment. Conversely, network-wide actions like the addition of users or changes to user policies must wait until the central control plane becomes available again. Magma makes trade-offs for availability versus consistency as the CAP theorem [22, 28] implies. It is generally possible for state stored in an AGW to be stale during times of disconnection, which might, for example, allow a UE to temporarily consume resources beyond its quota.

This design helps to achieve the scaling goals of Magma, in allowing both a small minimum footprint (scaling down) as well as scaling up. A minimal Magma deployment would be a single AGW and an orchestrator. The orchestrator is typically three virtual machine instances in a cloud, while the AGW itself is a small (4-core) x86 commodity server. This is dramatically less hardware than a conventional cellular packet core. Scaling up is essentially a matter of adding more AGWs, which increases the number of base stations and UEs, without much increase in the load on the orchestrator. We discuss our experiences in scaling up in Section 4.3.

The decision to place local control-plane functions on the

AGWs, while beneficial for scalability, does introduce trade-offs. In particular, it complicates the picture for some features that require coordination among AGWs. Notably, while Magma supports mobility across radios served by a common AGW, seamless mobility *between* AGWs would require communicating some control-plane state from one AGW to another during hand-offs. While many use cases can be supported without this feature, we expect to add it in the future.

3.3 Fault Tolerance Via Small Fault Domains

The desire to build a low-cost solution for Magma has a significant effect on how the architecture approaches fault tolerance. Low-cost hardware is prone to failure, and so Magma adopts the view common to most modern cloud systems: it is expected that individual components will fail. A failure of a component must affect as few users as possible (i.e., fault domains must be small) and must not affect other components. This approach also has a positive impact on operations such as software upgrades, as it is possible to upgrade small components independently without taking down the whole system. This is in stark contrast to traditional 3GPP implementations.

The SDN-like architecture of Magma localizes state more fully than a typical 3GPP implementation. In a standard implementation, the runtime state of a UE is spread among several large components (e.g., the PGW, SGW, and MME in the LTE case). In contrast, Magma localizes the runtime state of a UE to a single AGW. This simplifies failure handling. The runtime state stored in an AGW is checkpointed regularly and may be copied to a backup instance of the AGW running as a cloud service. When an AGW fails, the backup cloud instance is brought into service, and can manage connections for the affected set of UEs until the primary AGW is restarted. As noted above, an AGW may continue to establish sessions to UEs even when disconnected from the orchestrator. The state synchronization approach described in Section 3.4 mitigates the long-term effects of such failures.

While it is common for a traditional cellular packet core to serve millions of subscribers, Magma distributes much of the functionality to a large number of Access Gateways. Each AGW is thus a fault domain that holds state for a relatively small number of UEs served by a small number (typically less than ten) of base stations. The failure of a single AGW would impact the set of UEs currently served by the attached base stations, but has no impact on the rest of the network or its customers. This contrasts with the much larger fault domains typical of a standard mobile core implementation.

3.4 State Synchronization

State in a mobile core needs to be communicated among components. Generally, one component is the authoritative owner of some piece of state, and it needs to synchronize state

with another component. In Magma, state can take one of three forms, for which Magma makes different guarantees.

The first is *runtime state* associated with a UE and its network activity. Backwards compatibility with existing user devices and RAN equipment requires Magma to implement standards-defined state machines to support operations like connecting to the network; modifications to runtime state can occur due to events in the UE itself, the RAN equipment, or Magma's "core" network elements. Importantly, runtime state within Magma is encapsulated within the AGW, which as discussed in Section 3.3 is the failure domain for Magma, and we assume a crash-recovery failure model for AGWs.[1] Further, most runtime state is both ephemeral and recoverable in the event of failure: a UE can simply reconnect.

The second is the *configuration state*, associated with the configuration of a Magma network element, such as an AGW. This is only ever written by the orchestrator and pushed asynchronously to the AGW. Examples include classes of network policy to be applied to classes of user or radio configuration to be applied by an AGW to connected RAN equipment. AGWs recover configuration state after a crash, and the source of truth for configuration state is stored durably in the orchestrator (Postgres); we only permit modification to configuration state through the orchestrator. Configuration state generally changes on human timescales (i.e., minutes or hours).

Finally, Magma also manages *metrics state*, telemetry from Magma elements. This operational data, while useful, is captured on a best-effort basis.

Like many cloud-native systems, Magma adopts a "desired state" model for runtime and configuration state. By this we mean that to communicate a required state change (e.g., the addition of a new session in the data plane), the desired end state is set via an API. This contrasts with a "CRUD (Create, Read, Update, Delete)" interface, which is common in 3GPP specifications. Magma replaces the CRUD model with the desired state model to simplify reasoning about changes across elements of the system in the case of partial failures. This is a common case in challenged environments, where portions of the end-to-end system (e.g., backhaul) are far less reliable than others (e.g., the link between the UE and the RAN). This is best explained via a simple example.

Consider the case of establishing data-plane state in an AGW for a set of active sessions. Suppose there are two active sessions, X and Y. Then a third UE becomes active and a session Z needs to be established. In the CRUD model, the control plane would instruct the data plane "add session Z". The desired state model, by contrast, communicates the entire new state: "the set of sessions is now X, Y, Z". The CRUD model is brittle in the face of failures. If a message is lost, or a component is temporarily unable to receive updates, the receiver falls out of sync with the sender. So it is possible that the control plane believes that sessions X, Y and Z have

been established, while the data plane only has state for X and Y. By sending the entire desired state, we ensure that the receiver comes back into sync with the sender once it is able to receive messages again.

This approach is hardly a novel idea in the cloud-native world, but it differs from typical 3GPP systems. It allows Magma to tolerate occasional communication failures (caused by poor quality backhaul, for example) or component outages due to software restarts, hardware failures, etc. Limiting the scope of 3GPP protocols to the very edge of the network gave us the flexibility to rethink state synchronization to improve fault tolerance (in addition to other benefits noted above).

We close by considering how Magma manages state for one particularly salient policy: billing users based on data volume, and the possibility of double-spending. Volume-based billing policies are typically implemented using a third-party online charging system (OCS) that integrates with both the network operator's existing business support systems (BSS) as well as Magma. In this arrangement, *billing and charging* are handled by the OCS, while Magma handles *metering and accounting*. The OCS tracks a user's account balance (e.g., in US$) and then authorizes small quotas of data (e.g., 1MB) to the user via Magma; when the user nears completion of their quota, Magma requests another quota on the user's behalf from the OCS, which makes the decision on whether to grant or deny the request. Whether or not a user has been allocated a quota is *configuration* state from Magma's perspective, while the amount remaining in a user's current quota is *runtime* state. Thus, while it is possible for a malicious user to double-spend by moving between AGWs strategically, the maximum amount of double-spend permitted is capped as a business decision by the quota size. Operators for whom this is a particular concern could also adopt techniques for volume-based accounting in a distributed context [31].

3.5 Software Data-Plane Implementation

The data plane is responsible for (i) recognizing the flows for active sessions (traffic to and from active UEs); (ii) collecting statistics for those flows; (iii) adding and removing tunnel headers; and (iv) enforcing policies such as rate limits per subscriber. Magma's data plane is implemented using Open vSwitch (OVS) [47]. OVS provides a programmable data plane that is controlled by OpenFlow [39]. While OpenFlow and OVS are convenient implementation choices, they are not fundamental to the architecture. Other options may be used in the future. The important points are that the data plane is highly programmable and implemented entirely in software.

The software implementation of the data plane enables Magma to operate on commodity hardware. While throughput, latency, and jitter of the data plane are important for cellular networks, we have found OVS to offer entirely adequate performance. OVS performance has been well studied and optimized for many years [47]. In Section 4 we evaluate

---

[1]We generally assume the same for individual AGW software components; per-process state is held externally for most critical services.

the performance of OVS in the Magma context. It is worth noting that other aspects of the system such as backhaul and RAN capacity are likely to have a larger performance impact overall than the data plane within the access gateway.

The "data plane configuration" box in Figure 4 generates the commands necessary to program the data plane with a set of rules to handle the flows of current sessions. Currently, those commands are OpenFlow commands. If OVS were replaced with a different forwarding engine, only the "data plane configuration" component would be affected.

3.6 Federation With Other Networks

To this point we have described standalone deployment of Magma, but it can be deployed in one of three modes:

- *Standalone*: Magma supports an independent network, with all 3GPP control and user plane traffic terminated in the AGW.
- *Local breakout roaming*: Magma *federates* with an existing cellular network, with control-plane traffic terminated externally but user-plane traffic still handled by the AGW and routed directly to or from the Internet.
- *Home roaming*: Magma federates with an existing cellular network, with both control and user-plane traffic terminated in an external network.

Much as the AGW terminates access-specific protocols from the radio network, Magma introduces additional elements to terminate access-specific protocols with an external core network, using a component referred to as a *Federation Gateway* (FeG). The FeG implements 3GPP-defined interfaces to support "home roaming" as well as "local breakout roaming". The latter is made possible in Magma by the fact that rich policy enforcement is provided in the AGW. As an example, an AGW can obtain the policy to apply to a UE by querying the subscriber data base in the federated network, then enforce that policy in the AGW. Signalling traffic between UEs and the MNO core is handled by the FeG service in the orchestrator[2]. User data-plane traffic is tunneled to an analogous component, the GTP Aggregator (GTP-A) which in turn connects to the MNO's existing P-GW.

Unlike the AGW, the FeG and GTP-A are centralized, on-path devices. This serves a practical purpose: traditional MNOs prefer a single point of interconnection between their sensitive core network and "extension" networks [31]. This has scaling implications as discussed in Section 4.3.2.

4 Evaluation

Magma makes a number of fundamental design choices that differ from traditional core network software to improve flexibility and scalability, while supporting rich network policies.

The aim is to support *practical* cellular access deployments. To evaluate Magma, we first consider system performance in an emulated environment, and then discuss a large-scale commercial Magma deployment.

4.1 Supporting Typical Deployments

Emulation Testbed Although evaluating Magma's performance in a real deployment is possible at small scale, evaluating scenarios with hundreds of UEs and RAN elements is impractical. Further, extracting data from commercial deployments is challenging due to privacy and commercial considerations. Thus, we instead evaluate Magma using a commercial emulation system, Spirent Landslide [53], which allows us to emulate arbitrary configurations of virtual UEs and RAN elements in a replicable fashion.

For our evaluation, we deployed the most recent stable release of Magma, v1.6.1. We deployed the orchestrator on a cluster of AWS EC2 instances and two AGWs in our lab. The first AGW was a bare-metal AGW on an Intel J3160 quad-core 1.6GHz CPU with 8GB of RAM and four Intel I210 1Gbps NICs. The second was a virtual AGW running with Intel Xeon 6126 2.60GHz, 8GB of RAM, and 2x10G Mellanox ConnectX-3 NICs; we assigned a variable number of vCPUs to the virtual AGW as defined in our experiments below. Both the bare-metal and virtual AGWs were connected directly to the Landslide emulator as well as to the Internet via 1Gbps and 10Gbps links, respectively. We also verified that memory was not a bottleneck for the AGW during our experiments and that all machines in the orchestrator deployment were running well under capacity. Finally, the emulated SIM cards for the emulated UEs were pre-provisioned into the orchestrator and AGW in advance of all experiments, as is typical for network operator deployments of Magma.

Unlike traditional core networks, Magma's AGW is co-located with RAN equipment (for example, at a tower site), and the unit of scaling for Magma is the AGW itself: as operators grow their network, they add both additional RAN capacity (i.e., radio equipment) but also additional "core" capacity (i.e., AGW instances). Since the AGW is an on-path device for all traffic associated with the cell site, the AGW should be provisioned such that site is limited by the capacity of the RAN as the site, rather than the AGW. This is a notable observation that, in part, motivates Magma's design: when co-locating core network functionality with RAN elements, the *RAN is the bottleneck* for performance on a per-site basis.

The recommended (and typical) deployment scenario has roughly one AGW per "cell site", which in practice consists of 1-3 eNodeBs in the case of an LTE network. A typical eNodeB (such as those described in Table 2 or depicted in Figure 2) can support at most 96 simultaneously active users[3] and radio channels of at most 20MHz; this channel capacity, in turn, corresponds to a peak aggregate throughput of

---

[2]This is necessary to coordinate low-level network state between the UE and the MNO's traditional core, such as GTP bearer identifiers.

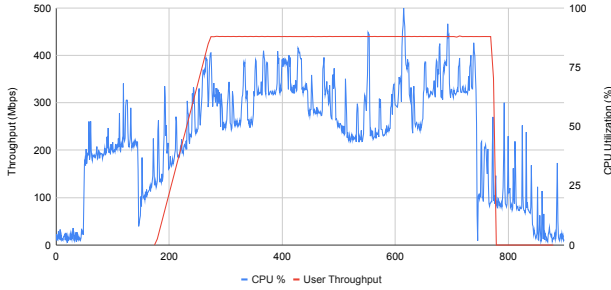[3]More users may be attached but not actively transmitting data.

Figure 5: AGW CPU utilization under maximum "typical" workload for a cell site. Aggregate throughput is limited by radio capacity, not the AGW.

Item	Unit Cost	Qty	Total	Notes
LTE eNodeB	US$4,000	3	US$12,000	Baicells Nova 223: 1W, 3.5GHz, 96 user, 2x2 MIMO.
AGW	US$450	1	US$450	Same as used in experiments.
Accessories	US$450	3	US$1,350	18dBi sector antenna, RF cables, connectors, grounding.
RAN CapEx (per site)			US$18,760	

Table 2: Cost breakdown of active RAN equipment for a typical Magma deployment. Excludes site-specific passive infrastructure and backhaul costs.

126Mbps [16] under ideal conditions, for a typical cell site maximum capacity of 378Mbps. We note that the additional cost of an AGW is modest in comparison to the cost of a cell site, similar to the site cost breakdown observed in related work [31]. Although LTE site costs can vary widely and are, in our experience, dominated by non-networking costs such as land, power, and tower (also known as "passive infrastructure"), a representative deployment could consist of the hardware in Table 2; AGW cost represents less than 3% of the cost of active equipment for the site. Power costs can be especially significant in "off-grid" locations, but these are largely driven by the power needs of the radio equipment and hence not greatly influenced by the mobile core implementation. Note the use of solar and battery power in Figure 2.

Magma must be able to support this type of workload. We evaluate this by emulating the peak load of a the cell site described above: a total of 288 UEs connect (or "attach") to the network for the first time at a rate of 3UE/sec, and each then performs a short HTTP download at a rate of 1.5Mbps, for an aggregate total offered load of 432Mbps. Figure 5 demonstrates our results, focusing on the total CPU utilization as well as achieved throughput of the AGW. At a high level, the AGW accepts attach requests from all new users over the course of approximately 1.5 minutes, after which the AGW enters a steady state for the duration that UEs are making HTTP requests. In this experiment, average sustained UE throughput reaches the expected throughput of 432Mbps throughout the duration of the experiment, indicating performance is limited by the RAN, rather than Magma's AGW, as expected.

We acknowledge that other RAN configurations can exist (including vRAN/cRAN arrangements) where many RAN ele-
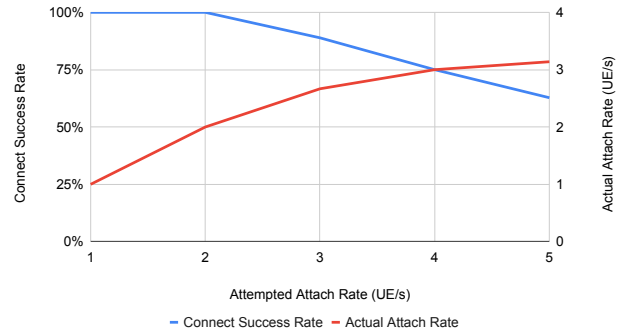

Figure 6: Maximum supported attach rates are limited by the AGW (specifically, the MME component). Results depict a physical AGW.

ments are effectively "co-located" to a single point within the operator's network. Magma can be used effectively in these deployments, with one (or more) AGWs allocated to support this range of RAN equipment. However, no deployments at scale of Magma to date have used that configuration (to our knowledge), and Magma can be deployed on any general-purpose compute (e.g., VM or container) alongside this RAN infrastructure to support it. Similarly, a radio vendor could integrate an AGW into the same physical enclosure as a traditional eNodeB for a combined RAN and AGW element.

4.2 Control and User Plane Separation

Different usage patterns of a network stress user plane or control plane elements of the network core: a common example of the former would be human users accessing video content while the latter would be an IoT workload consisting of large numbers of devices that only exchange occasional small messages. This presents a major dimensioning challenge in traditional cellular core networks and motivates efforts to separate control and data plane elements so operators can scale them independently (statically or dynamically); this is known as "control/user plane separation" (CUPS) in LTE and 5G.

Magma's distributed design naturally facilitates a CUPS architecture. By default, every AGW implements a data plane at the network edge, and all control plane functions are implemented as user-space processes at the AGW, with configuration state managed by the orchestrator.

From Figure 5, we observe that the AGW operates in two distinct and characteristic domains. At the start of our experiment, while UEs are attaching to the network, the AGW's CPU workload is dominated by the *control plane* workload associated with handling attach requests, including performing cryptographic operations necessary to authenticate users as well as setting up per-user, per-session state in the data plane and control plane to implement the desired policy for each UE; in our experience, this is the most computationally-intensive control plane procedure. After UEs attach, the CPU workload is dominated by *user plane* workload associated with forwarding UE traffic.
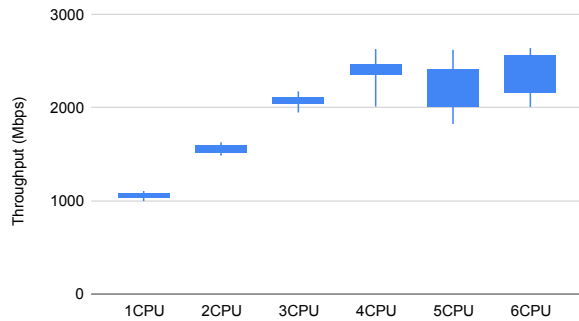
Figure 7: Steady state throughput vs. CPUs allocated to user plane. Note our traffic generator was unable to saturate the virtual AGW's user plane in the 5CPU case and above.

Figure 6 illustrates how our bare-metal AGW copes with a "worst case" control plane workload, a surge of new UEs attaching then saturating the data plane. We define the *connection success rate* (CSR) to be the number of connection attempts that succeed over the total number of connection attempts made, for each five second bin during the experiment. We observe that above 2UE/s, the bare-metal AGW is unable to service all connection attempts, with the connection success rate (CSR) falling linearly beyond this point. On a per-AGW basis, Magma's control-plane performance is relatively limited; improving this is an active area of engineering effort. Attach rate is a function of hardware as well: a 4 vCPU instance of our virtual AGW supports 16 attaches per second, which would saturate the RAN capacity of the "typical" site described above in 18 seconds.

Lastly, we consider per-AGW allocation of resources to the control and user plane. To do this, we statically limit the number of cores available to the user plane and evaluate steady-state throughput and median connection success rate. These results are shown in Figures 7 and 8; note that these experiments use the VM AGW, and as such the absolute throughput numbers are not comparable with earlier experiments. We observe that increasing the cores available to the user plane improves steady-state throughput at the cost of decreased connection success rate (i.e., control-plane performance), but allowing the kernel scheduler to allocate resources flexibly between user plane and control plane tasks provides both high throughput and good connection success rates. We note that we expect raw user-plane performance to increase beyond what is shown here; the commercial test equipment we used was unable to generate more than 2.5Gbps aggregate load.

Taken together, these emulation results demonstrate that Magma can handle typical workloads using low-cost commodity hardware. For more intensive workloads, Magma's control and user plane capacity scales with additional hardware. We finally note that these results provide an upper-bound on the performance of a *single* Magma AGW; the *network* capacity of a Magma network scales linearly with AGWs.
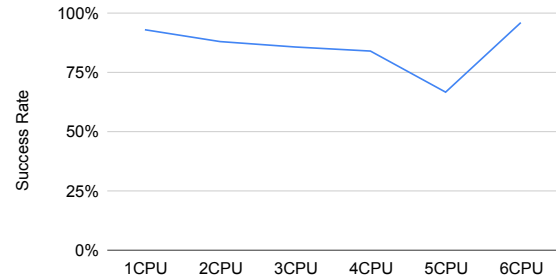

Figure 8: Median connection success rate vs CPUs allocated to user plane.

4.3 Deployment

We now turn to large commercial deployments of Magma. We first note that Magma is an open-source project governed by the Linux Foundation, and as such the core development team (including the authors of this paper) do not directly operate any production deployments; as such, we draw our examples from partners within the project's ecosystem.

Magma adoption. To understand how Magma is used in practice, we interviewed two people working in product management and marketing for the Magma open-source project; in their roles, they speak regularly with operators as well as other commercial entities within the Magma ecosystem. Based on our discussion, as of February 2022, twenty commercial networks were operating using Magma across eight countries in Africa, Asia, North America, and South America. These networks support a range of access modalities and policies. For example, Magma has been used in networks providing backhaul for WiFi hotspots, fixed wireless broadband to homes and businesses, "carrier" WiFi to extend a traditional mobile operator's service to indoor WiFi, and traditional mobile broadband service. Today, Magma has approximately 100 active committers to its codebase.

Magma deployments. To demonstrate how Magma is used, we worked with one of the largest commercial entities, FreedomFi, that provides support to operators deploying Magma. FreedomFi provided data to characterize two significant deployments they help operate. This data was provided to the authors in de-identified form, and only operational data (not user data) was used in our analysis.

4.3.1 Fixed Wireless Hotspots

One of FreedomFi's first commercial deployments was AccessParks [1], a US-based operator that provides public WiFi hotspot networks in large outdoor areas; their deployment locations require multiple WiFi access points (APs) to provide consistent service. With the availability of CBRS spectrum, AccessParks sought to use LTE to provide backhaul to their WiFi hotspots in some of their larger deployments. End users connect to AccessParks's WiFi access points via traditional WiFi mechanisms and an existing captive portal system, and
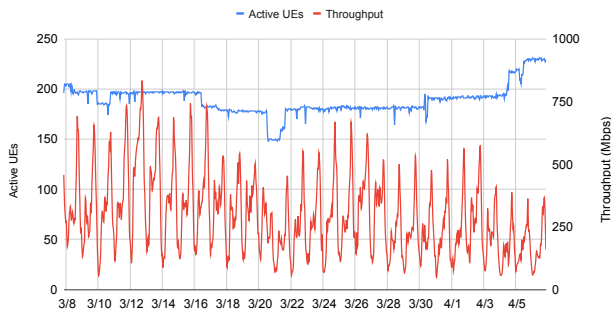
Figure 9: Per-hour AccessParks usage during Mar-Apr 2022.

Item	Traditional	Magma	Difference (%)	Notes
RAN	$7,950	$7,950	-	Identical RAN and backup power.
Core HW	$1,200	$300	-$900 (-75%)	
Core SW	$2,000	$600	-$1,400 (-70%)	Licenses/support.
Field Eng.	$200	$200	-	Installation.
LTE Eng.	$5,000	$330	-$4,670 (-93%)	Planning, core config.
Cost/Site	$16,350	$9,380	-$6,970 (-43%)	

Table 3: Comparison of per-site installed costs for Access-Parks's traditional cellular system compared to Magma. Total cost per site decreased by 43%, driven primarily by Magma's reduction in operational complexity for deployment.

the UEs in the Magma network are fixed wireless modems that connect the WiFi APs to the Internet via Magma. The setup is illustrated in Figure 10.

AccessParks's deployment began in December 2020 with a ten site pilot to evaluate Magma. Today, the network consists of fourteen sites providing backhaul to over 200 access points, with plans to continue expanding. Figure 9 depicts active subscribers and hourly throughput of the network.

Network policies for the AccessParks networks are very simple: because the LTE network simply serves as backhaul, all UEs simply have unrestricted access. Per-user policies are implemented by AccessParks's pre-existing captive portal and pre-paid billing software, which is implemented using standard techniques (i.e., RADIUS for AAA at the WiFi AP).

Operational complexity. AccessParks's original Magma pilot was motivated in part by their poor experiences with the operational complexity of other commercial and open-source cellular core software in their previous two years of deployment. Although operational complexity is subjective, one quantifiable way in which it manifests is in an operator's labor costs: simpler systems should require less staff time and support to manage. Table 3 shows the results of this comparison for AccessParks. For identical access network infrastructure, AccessParks achieved a 43% reduction in per-site deployment costs using Magma compared to traditional architectures, largely driven by a reduction in support costs and engineering time for site configuration and planning.[4]

4.3.2   Franchised MNO Extension

A second (and, to our knowledge, the largest) deployment of Magma is an early-stage deployment to provide a franchised, neutral host network.[5] This network is unique in that the physical deployment of network infrastructure is not man-

aged by any single network operator. Instead, "micro network operators" (which include individuals, small ISPs, and enterprises) deploy LTE and 5G RAN equipment alongside Magma AGWs that have been customized by FreedomFi to support their proprietary traffic accounting and settlement system.

Services and Policy. The neutral host network is operated by FreedomFi and allows customers of incumbent MNOs to use this network for service. The core "policy" supported by this network is tunnelling all user traffic back to the appropriate MNO; a user's MNO, in turn, applies their standard network policies for billing, charging, and throttling within their existing core network. The FreedomFi network provides access on a best-effort basis, with each micro network operator leveraging shared CBRS [17] spectrum in the 3.5GHz band (as done in the previous deployment). This service requires integrating the thousands of distributed AGWs with a partner MNO's centralized core network, leveraging the federation capabilities described in Section 3.6.

Scale. As of this writing, this network is still in early testing, so does not have significant user traffic. However, it still provides a useful example of how the Magma *control plane* scales with network size: even without users, Magma still manages device configuration, network monitoring, and supports interconnection with partner MNO core networks.

The FreedomFi network began initial deployments in November 2021, and as of April 2022 consists of 5370 AGWs and 880 eNodeBs (FreedomFi reports the discrepancy between AGWs and eNodeBs is due to supply-chain issues: while AGWs are commodity x86 PCs, cellular radios are specialized equipment with fewer vendors and the ones used in this network only began shipping in January 2022). The network is currently adding on average 150 new AGWs and 90 new eNodeBs per week, all of which are deployed on an ad-hoc basis by micro-network operators; these AGWs are deployed in 45 states across the United States.[6]

Supporting this network is a dedicated orchestrator running on six AWS virtual machines managed by Kubernetes (EKS) [18]. Three instances are dedicated towards "heavy" tasks: operation of the FeG, device configuration, and metrics reporting; these systems are each equipped with 16 vCPUs and 32GB RAM. Remaining orchestrator services run on a

---

[4]Unfortunately, we do not have data on ongoing maintenance costs from AccessParks; however, AccessParks' decision to use Magma for future deployments suggests it compared favorably.

[5]A neutral host network describes a business model in which a mobile network is operated by an entity for the sole purpose of providing wholesale capacity to third-party retail MNOs and MVNOs; the neutral host network operator does not have its own users, but instead enables users of its customers to use the neutral host network on a shared basis.

[6]The network only operates in the United States for regulatory reasons.

collection of smaller VMs (4 vCPU/16GB RAM). The GTP-A runs on a single bare metal server with a 3.4GHz 8-core Xeon E2278G CPU, 32GB RAM, and 2x10G NICs, and is physically co-located near the facilities of a partner MNO's core network. In total, this costs FreedomFi approximately $4,000 per month to operate.

We view the rapid deployment of this network as cautious evidence for Magma's ability to support large-scale networks with unique business models. We hope to further investigate the operational dynamics of this network in future work.

5 Related Work

Open-source LTE/5G core networks: Several projects share our goal of creating an open-source LTE/5G cellular core network [4, 9, 10, 13]; these were preceded by similar efforts to build open 2G and 3G networks [11, 12]. With the exception of OpenBTS [11] (a GSM-to-VOIP bridge), each of these focuses on implementing traditional, 3GPP-compliant, core networks.[7] Aether [2, 43] is an open-source 5G-connected edge platform, which brings together 5G connectivity and edge-cloud servers. Like Magma, Aether adopts cloud design principles. However, Aether does not refactor the network design to break the coupling of the radio access technology with the core, and Aether does not focus on low-cost equipment to reach under-served users.

Expanding connectivity access: Many efforts have proposed or described novel solutions for expanding Internet access to under-served people [25, 30, 44, 45, 48, 55, 56]. Similarly, small(er)-scale network operators have a rich history providing service to especially rural communities [27], such as community networks [3, 6, 7, 21] and small ISPs [32]. Of this extensive literature, Magma is most closely related to work on community cellular networks [19, 33, 51].

NextG cellular core architecture. The networking research community is actively rethinking the design of next generation networks. PEPC [50] refactors the packet core by consolidating user state into one location, similar in spirit to Magma's AGW. ECHO [40] refactors an EPC to run on less-reliable public cloud infrastructure. SCALE [20] explores an elastically scalable cellular control plane, and KLEIN [49] describes a similarly elastic control and data plane. Although these works all focus on (logically) centralized core networks, the techniques described are complementary to Magma.

Other work takes a more "clean slate" approach to reimagining the cellular core. CellBricks [38] contemplates a highly federated cellular network and moves support for mobility, authentication, and billing into end hosts; it is implemented as an extension to Magma. dLTE [35] makes 4G networks more like WiFi through a decentralized design, including a global registry for peer discovery. SoftCell [34] uses SDN

principles to improve the scalability and flexibility of the packet core network. Magma draws on this body of work for inspiration while maintaining a backwards-compatible, standards-compliant edge to facilitate production deployment.

Magma directly builds on recent work exploring core architectures for under-served communities. CCM [31] presents a distributed cellular 2G core that enables semi-disconnected operation over unreliable rural backhaul connections; this work served as an early inspiration for Magma, which extends these concepts to modern wireless access technologies. Similarly, CoLTE [52] provides a lightweight core which—like an AGW—is co-located with RAN elements, but unlike Magma focuses on small, independent community networks.

Open radio access networks: Several recent initiatives focus on opening up the radio access network (RAN). For example, the OpenRAN project [59] and the O-RAN alliance [8, 41] develops standards that disaggregate 3GPP RANs, with open interfaces between the layers. These efforts are complementary to Magma, as they focus on the cellular interface—the part of the network *before* reaching Magma's access gateway.

6 Conclusion

We have presented our experiences in designing and deploying Magma, an open-source platform for building access networks. The most important design decision was to terminate the RAN-specific protocols in access gateways close to the radio. This simple design decision brings many benefits: supporting diverse radio technologies, tolerating disruptions in backhaul links, using a low-cost software data plane, and scaling naturally with a hierarchical SDN control plane. Magma also adopts modern cloud-computing design patterns (e.g., desired-state synchronization, tolerance to failure of individual components) and open-source software components (e.g., gRPC, Open vSwitch, Kubernetes, Prometheus). In line with Magma's goal to enable practical networks, we demonstrated that Magma can support typical deployment scenarios and discussed two large-scale commercial networks that use Magma. Importantly, Magma also scales *down*, with a small minimum footprint that supports incremental deployment, thus filling a gap between traditional WiFi and cellular. All software artifacts for Magma are available on GitHub[8].

Magma was designed with the primary goal of reaching under-served communities, by supporting heterogeneous radio and backhaul technologies and reducing capital and operational cost. We believe that Magma is a good fit for other deployment scenarios, including enterprise 5G networks. Future work on Magma can expand the set of supported features, including seamless mobility between access gateways as well as network virtualization. We look forward to extending the Magma code base, and the community of contributors to the software, so the platform can evolve to serve more users.

---

[7]We note that the Magma AGW's LTE-specific portion was originally based upon OpenAirInterface [9], as it was the most mature open-source core available at the inception of Magma's development.

[8]https://github.com/magma/magma

References

[1] AccessParks. https://accessparks.com/. Retrieved 7/2022.

[2] Aether. https://www.aetherproject.org/. Retrieved 7/2022.

[3] AlterMundi. https://altermundi.net/. Retrieved 7/2022.

[4] free5GC. https://www.free5gc.org/.

[5] gRPC. https://grpc.io/. Retrieved 7/2022.

[6] Guifi.net. https://guifi.net/. Retrieved 7/2022.

[7] NYCMesh. https://www.nycmesh.net/. Retrieved 7/2022.

[8] O-RAN Alliance. https://www.o-ran.org/.

[9] Open Air Interface. https://openairinterface.org/.

[10] Open5Gs. https://open5gs.org/. Retrieved 7/2022.

[11] OpenBTS. http://openbts.org. Retrieved 7/2022.

[12] Osmocom. https://osmocom.org/projects/cellular-infrastructure. Retrieved 7/2022.

[13] srsRAN. https://www.srsran.com/.

[14] Universal Mobile Telecommunications System (UMTS); LTE; Policy and Charging Control (PCC); Reference points. https://www.etsi.org/deliver/etsi_TS/129200_129299/129212/15.03.00_60/ts_129212v150300p.pdf. (3GPP TS 29.212 version 15.3.0 Release 15).

[15] 3GPP: The mobile broadband standard. https://www.3gpp.org/.

[16] 3GPP: Evolved universal terrestrial radio access (E-UTRA); physical layer procedures. https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2427.

[17] CBRS Alliance. https://www.cbrsalliance.org.

[18] Amazon Elastic Kubernetes Service (EKS). https://aws.amazon.com/eks/.

[19] Abhinav Anand, Veljko Pejovic, Elizabeth M Belding, and David L Johnson. VillageCell: Cost Effective Cellular Connectivity in Rural Areas. In *International Conference on Information and Communication Technologies and Development*, pages 180–189, 2012.

[20] Arijit Banerjee, Rajesh Mahindra, Karthik Sundaresan, Sneha Kasera, Kobus Van der Merwe, and Sampath Rangarajan. Scaling the LTE Control-Plane for Future Mobile Access. In *ACM SIGCOMM CoNEXT Conference*, pages 1–13. ACM, 2015.

[21] Luca Belli, Sarbani Banerjee Belur, Peter Bloom, Anriette Esterhuysen, Nathalia Foditsch, Maureen Hernandez, Erik Huerta, Mike Jensen, Meghna Khaturia, Michael J Oghia, et al. *Community Networks: The Internet by the People, for the People*. FGV Direito Rio, December 2017.

[22] Eric Brewer. CAP Twelve Years Later: How The "Rules" Have Changed. *Computer*, 45(2):23–29, 2012.

[23] Jyh-Cheng Chen, Ming-Chia Jiang, and Yi-wen Liu. Wireless LAN security and IEEE 802.11i. *IEEE Wireless Communications*, pages 27–36, February 2005.

[24] T. Charles Clancy. Secure handover in enterprise WLANs: Capwap, Hokey, and IEEE 802.11R. *IEEE Wireless Communications*, pages 80–85, October 2008.

[25] Michaelanne Dye, David Nemer, Josiah Mangiameli, Amy S Bruckman, and Neha Kumar. El Paquete Semanal: The Week's Internet in Havana. In *CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018.

[26] Armando Fox, Steven D Gribble, Yatin Chawathe, Eric A Brewer, and Paul Gauthier. Cluster-based Scalable Network Services. In *ACM Symposium on Operating Systems Principles*, pages 78–91, 1997.

[27] Hernan Galperin and François Bar. The Microtelco Opportunity: Evidence from Latin America. *Information Technologies and International Development*, 3(2), 2006.

[28] Seth Gilbert and Nancy Lynch. Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services. *SIGACT News*, 33(2):51–59, jun 2002.

[29] GSMA. Unlocking Rural Coverage: Enablers for commercially sustainable mobile network expansion, 7 2016.

https://www.gsma.com/mobilefordevelopment/resources/unlocking-rural-coverage-enablers-commercially-sustainable-mobile-network-expansion/.

[30] S. Guo, M. H. Falaki, E. A. Oliver, E. A. Oliver, S. Ur Rahman, S. Ur Rahman, A. Seth, M. A. Zaharia, and S. Keshav. Very Low-Cost Internet Access Using KioskNet. *ACM SIGCOMM Computer Communication Review*, 37(5):95–100, 2007.

[31] Shaddi Hasan, Mary Claire Barela, Matthew Johnson, Eric Brewer, and Kurtis Heimerl. Scaling Community Cellular Networks with CommunityCellularManager. In *USENIX Symposium on Networked Systems Design and Implementation*, pages 735–750, 2019.

[32] Shaddi Hasan, Yahel Ben-David, Max Bittman, and Barath Raghavan. The Challenges of Scaling WISPs. In *Annual Symposium on Computing for Development*, pages 3–11, 2015.

[33] Kurtis Heimerl, Shaddi Hasan, Kashif Ali, Eric Brewer, and Tapan Parikh. Local, Sustainable, Small-Scale Cellular Networks. In *International Conference on Information and Communication Technologies and Development*, ICTD '13, pages 2–12, Cape Town, South Africa, 2013. ACM.

[34] Xin Jin, Li Erran Li, Laurent Vanbever, and Jennifer Rexford. SoftCell: Scalable and Flexible Cellular Core Network Architecture. In *ACM SIGCOMM CoNEXT Conference*, pages 163–174. ACM, 2013.

[35] Matthew Johnson, Spencer Sevilla, Esther Jang, and Kurtis Heimerl. dLTE: Building a more WiFi-like Cellular Network (Instead of the Other Way Around). In *ACM Workshop on Hot Topics in Networks*, pages 8–14. ACM, 2018.

[36] Matthew William Johnson, Esther Han Beol Jang, Frankie O'Rourke, Rachel Ye, and Kurtis Heimerl. Network Capacity as Common Pool Resource: Community-Based Congestion Management in a Community Network. *ACM Human-Computer Interaction*, 5(CSCW1):1–25, 2021.

[37] Teemu Koponen, Keith Amidon, Peter Balland, Martin Casado, Anupam Chanda, et al. Network Virtualization in Multi-tenant Datacenters. In *USENIX Symposium on Networked Systems Design and Implementation*, pages 203–216, 2014.

[38] Zhihong Luo, Silvery Fu, Mark Theis, Shaddi Hasan, Sylvia Ratnasamy, and Scott Shenker. Democratizing Cellular Access with CellBricks. In *ACM SIGCOMM*, August 2021.

[39] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[40] Binh Nguyen, Tian Zhang, Bozidar Radunovic, Ryan Stutsman, Thomas Karagiannis, Jakub Kocur, and Jacobus Van der Merwe. ECHO: A Reliable Distributed Cellular Core Network for Hyper-Scale Public Clouds. In *Annual International Conference on Mobile Computing and Networking*, pages 163–178. ACM, 2018.

[41] O-RAN Alliance. O-RAN: Towards an Open and Smart RAN, October 2018. White paper, https://www.o-ran.org/s/O-RAN-WP-FInal-181017.pdf.

[42] OVN Architecture. https://www.ovn.org/support/dist-docs/ovn-architecture.7.pdf, 2021.

[43] Guru Parulkar. Aether: An Open Source Platform for Private 5G Connected Edge Cloud-as-a-Service, 2020. Keynote presentation at ONF Spotlight on 5G Connected Edge Cloud for Industry 4.0 Transformation, https://www.youtube.com/watch?v=Zn7FZyiw5KM&t=4s.

[44] Rabin Patra, Sergiu Nedevschi, Sonesh Surana, Anmol Sheth, Lakshminarayanan Subramanian, and Eric Brewer. WiLDNet: Design and Implementation of High Performance WiFi Based Long Distance Networks. In *USENIX Symposium on Networked Systems Design and Implementation*, 2007.

[45] Alex (Sandy) Pentland, Richard Fletcher, and Amir Hasson. DakNet: Rethinking Connectivity in Developing Nations. *Computer*, 37(1):78–83, 2004.

[46] Larry Peterson and Oguz Sunay. *5G Mobile Networks: A Systems Approach*. Systems Approach, June 2020. https://5g.systemsapproach.org/.

[47] Ben Pfaff, Justin Pettit, Teemu Koponen, Ethan Jackson, Andy Zhou, Jarno Rajahalme, Jesse Gross, Alex Wang, Joe Stringer, Pravin Shelar, et al. The Design and Implementation of Open vSwitch. In *USENIX Symposium on Networked Systems Design and Implementation*, pages 117–130, 2015.

[48] Thomas Pötsch, Salman Yousaf, Barath Raghavan, and Jay Chen. Zyxt: A Network Planning Tool for Rural Wireless ISPs. In *ACM SIGCAS Conference on Computing and Sustainable Societies*, pages 1–11, 2018.

[49] Zafar Ayyub Qazi, Phani Krishna Penumarthi, Vyas Sekar, Vijay Gopalakrishnan, Kaustubh Joshi, and Samir R Das. KLEIN: A Minimally Disruptive Design for an Elastic Cellular Core. In *ACM Symposium on SDN Research*, pages 1–12. ACM, 2016.

[50] Zafar Ayyub Qazi, Melvin Walls, Aurojit Panda, Vyas Sekar, Sylvia Ratnasamy, and Scott Shenker. A High Performance Packet Core for Next Generation Cellular Networks. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 348–361. ACM, 2017.

[51] Rhizomatica. http://rhizomatica.org/, 2013. Retrieved 4/2013.

[52] Spencer Sevilla, Matthew Johnson, Pat Kosakanchit, Jenny Liang, and Kurtis Heimerl. Experiences: Design, Implementation, and Deployment of CoLTE, a Community LTE Solution. In *ACM MobiCom*, pages 1–16, 2019.

[53] Spirent Landslide Core Network Testing. https://www.spirent.com/products/mobile-network-testing.

[54] Statista. Internet usage worldwide—statistics & facts. https://www.statista.com/topics/1145/internet-usage-worldwide/#dossierKeyfigures.

[55] Sonesh Surana, Rabin K Patra, Sergiu Nedevschi, Manuel Ramos, Lakshminarayanan Subramanian, Yahel Ben-David, and Eric A Brewer. Beyond Pilots: Keeping Rural Wireless Networks Alive. In *NSDI*, volume 8, pages 119–132, 2008.

[56] William Waites, James Sweet, Roger Baig, Peter Buneman, Marwan Fayed, Gordon Hughes, Michael Fourman, and Richard Simmons. RemIX: A Distributed Internet Exchange for Remote and Rural Networks. In *Workshop on Global Access to the Internet for All*, pages 25–30, 2016.

[57] WBA. OpenRoaming. https://wballiance.com/openroaming/, 2021.

[58] Klaas Wierenga and Licia Florio. Eduroam: Past, Present and Future. *Computational Methods in Science and Technology*, 11(2):169–173, 2005.

[59] Mao Yang, Yong Li, Depeng Jin, Li Su, Shaowu Ma, and Lieguang Zeng. OpenRAN: A Software-Defined RAN Architecture via Virtualization. *ACM SIGCOMM Computer Communication Review*, 43(4):549–550, 2013.

[60] H. Zhu, M. Li, I. Chlamtac, and B. Prabhakaran. A Survey of Quality of Service in IEEE 802.11 Networks. *IEEE Wireless Communications*, pages 6–14, August 2004.
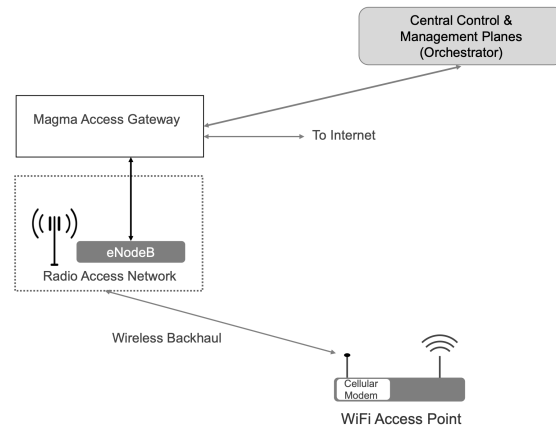
Appendix

Figure 10: Wireless backhaul to WiFi hotspots provided by Magma. This is the network architecture used by AccessParks in their deployment: end users connect to WiFi access points via standard mechanisms, and traffic is backhauled from the hotspot via a co-located cellular modem to the LTE RAN supported by Magma. Note that nothing in this design precludes an end user from directly connecting to the LTE network, if appropriately configured and allowed to do so by the network operator.

Acronym	Definition
MME	Mobility Management Entity
HSS	Home Subscriber Server
PCRF	Policy and Charging Rules Function
SGW	Serving Gateway
PGW	Packet Gateway
AMF	Access and Mobility Function
SMF	Session Management Function
PCF	Policy Control Function
UDM	Unified Data Management
AUSF	Authentication Server Function
S1AP	S1 Access Protocol
NGAP	Next Generation Access Protocol
SCTP	Stream Control Transmission Protocol
NAS	Non-Access Stratum
RAN	Radio Access Network
LTE	Long Term Evolution
3GPP	Third Generation Partnership Project
UE	User Equipment (a phone or other cellular client)
eNodeB	The "access point" for an LTE network
gNodeB	The "access point" for an 5G network
AGW	Access Gateway
AAA	Authentication, Authorization, and Accounting
RADIUS	Remote Authentication Dial-In User Service

Table 4: Acronyms used in the paper