Room-Area Networks

Peter A. Iannucci, Ravi Netravali, Ameesh K. Goyal, and Hari Balakrishnan
MIT CSAIL
{iannucci, ravinet, ameesh, hari}@mit.edu

Abstract

This paper makes the case for "Room-Area Networks" (RAN), a new category that falls between personal area networks and local area networks. In a RAN, a set of nodes can hear each other only if they are in the same room, broadly construed as being within earshot. We define a RAN abstraction, and we present example applications ranging from social contact management to building automation to gaming where this abstraction will help. The requirements of a RAN are poorly served by current technologies such as Bluetooth, near-field communication (NFC), Wi-Fi, and infrared. Acoustic channels, on the other hand, are well-suited in principle for effective propagation within human earshot and sharp attenuation at room boundaries. We provide a portable reference implementation of an 802.11a-like physical layer for the acoustic medium that works on current mobile devices, with successful communication even in noisy environments at distances over 8 meters.

CCS Concepts

•**Networks** → **Short-range networks; Mobile networks;** *Layering; Network mobility;* •**Hardware** → **Sound-based input / output;** •**Security and privacy** → *Multi-factor authentication;* Access control;

1. INTRODUCTION

An emerging class of mobile applications and services stands to benefit from room-level networking with zero configuration (Table 1). For example:

- Bump [16] was an application (with over 100 million downloads) for sharing contact information. It required access to the cloud and extremely close physical proximity. A network between devices in the same room would alleviate both requirements.
- Comfy [6] is an application for localized indoor heating, ventilation, and air conditioning control. It does not know which room the user is in, instead requiring the user to click on a floor plan. It cannot prevent a user in one room from altering the climate in another room.
- GameKit Session [2] was a library for nearby peer discovery for multi-player gaming on iOS. The room or space occupied by the human players becomes a proxy for a game lobby in which players coordinate and prepare to enter a shared gaming session.
- shopBeacon by Shopkick [24] grants shoppers rewards for visiting a retail establishment. When the user enters the physical space near a beacon, the application is notified to provide the user with marketing deals.
- Google Tone [18], Chirp [5], AirDrop [20], and Android Beam [1] provide functions related to file or link sharing. The physical range over which they operate varies: Beam requires near-field communication (NFC), Tone and Chirp require earshot, and AirDrop requires radio frequency (RF) contact. The key idea in each case is that multiple devices in the same space would like to find each other to coordinate a transfer, which may be through the wide-area network (WAN) or even the cloud.
- Room-level Wi-Fi auto-configuration (§3.1-3.2). Bob, a network administrator, would like to ensure that visitors to his company can get easy access to the secure wireless network without any manual exchange of passwords, but only if they are in certain rooms of the building. In addition, he would like to ensure that users in the rooms used for press demos and videoconferencing receive higher throughput than users located elsewhere when the network is loaded. He would also like to rate-limit the traffic produced by devices in the building's main lobby, which is open to the public, and to limit users in other rooms of the building to connect only to certain Wi-Fi access points (APs), even when other APs are visible.

These applications either explicitly adopt or stand to benefit from a network which follows the human notion of connectivity within earshot.

The purpose of this paper is to ask what abstraction best meets the needs of these applications, and to propose a suitable network architecture to implement it. We describe our room-area network (RAN) stack in §2. The key properties of a RAN, as we define them, are that it should

1. Adhere to room boundaries,
2. Work over a broadcast medium, and
3. Provide unicast, multicast, and service discovery.

Our solution has several additional favorable properties: it is

HotNets-XIV, November 16 - 17, 2015, Philadelphia, PA, USA

ACM 978-1-4503-4047-2/15/11...$15.00
DOI: http://dx.doi.org/10.1145/2834050.2834113

Domain	Application	
Multimedia	Waves by Wham City Lights	[21]
Retail & Marketing	shopBeacon by Shopkick	[24]
—"—	Dash Button by Amazon	
Home Theater	Chromecast Guest Mode by Google	[12]
Wi-Fi Configuration	Secure guest access	§3.1
—"—	Rate limits and traffic prioritization	§3.2
Building Automation	Room detection for comfort control	§3.3
Contact Management	Bump by Bump Technologies	[16]
Building Automation	Comfy / BOSS by UC Berkeley	[6, 9]
Gaming	GameKit Session by Apple	[2]
File/Link Sharing	Google Tone by Google	[18]
—"—	Chirp by Animal Systems	[5]
—"—	AirDrop by Apple	[20]
—"—	Android Beam by Google	[1]

Table 1: Applications of RAN. RAN fully captures the concepts expressed by the first group. The second group represents applications which stand to gain in usability from the RAN abstraction.

4. Not bound to any one physical layer,
5. Uses the familiar Berkeley sockets API,
6. Cross-platform, and
7. Supports any number of simultaneous applications.

In §3, we show how to exploit this architecture to support interesting and novel access control applications for Wi-Fi, building automation, and home theater. To date, these applications have been supported by approximations to the RAN concept, and have had to rely on their own custom solutions.

Ideally, an implementation of a RAN would use technologies available on current mobile devices. However, signals from radio-based systems tend to penetrate walls and doors, making them align poorly with the requirements for RAN. The best remaining choice for a RAN physical layer is the acoustic channel [4, 21, 22, 24]. We provide a reference implementation of an acoustic PHY based on 802.11a, which we call *Blurt*. Blurt has been carefully tuned to tolerate noise and interference, to be unobtrusive to humans, to be computationally efficient, to work over ranges of several meters, and to adapt to the non-idealities of the acoustic medium. We found that Blurt signals are attenuated by 4000× more than Wi-Fi, thus adhering much more intuitively to room boundaries (Table 2).

	Glass Door	Wooden Door
Blurt	36 dB	40 dB
Wi-Fi	1.5 dB	2 dB

Table 2: Attenuation of Wi-Fi and ultrasound through some typical office barriers.

2. NETWORK STACK

Our RAN stack is depicted in Fig. 1. Other physical layers besides Blurt are possible, including infrared (§4.2).

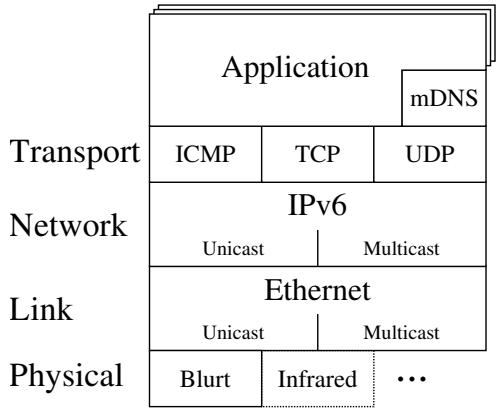


Figure 1: Layers in the proposed RAN stack.

2.1 Link Layer

A RAN link is a shared, possibly low-bandwidth Ethernet segment providing best-effort delivery. Each node applies carrier-sense multiple access (CSMA) and exponential-weighted random back-off.

Ideally, we will provision globally unique MAC addresses to RAN-equipped devices. If this is too onerous or expensive, one alternative is for each node to randomly choose a MAC address from the locally-administered unicast range. There are 2^{46} such addresses, giving a negligible probability of collision in a room containing less than 10000 devices.

At the link layer, unicast traffic is addressed directly to the destination node. Multicast relies on the broadcast nature of the medium to reach interested nodes. The software stack at the receiver is responsible for keeping track of which unicast and multicast addresses the interface will accept.

2.2 Network layer

Because the entire RAN is one shared segment at the link layer, nothing is to be gained from maintaining a distinction between MAC and IP addresses. If it is feasible to remove this distinction, then we will have no need for ARP, saving time for unicast connection establishment.[1] We will then also have no need for DHCP, since IP addresses will be computed from unique MAC addresses. Avoiding DHCP is highly desirable, since designating a DHCP server introduces an undesirable asymmetry between nodes in the RAN and raises thorny questions about what happens when the DHCP server leaves the room.

There are two reasons why we cannot achieve this goal with IPv4. First, IPv4 has only 65536 link-local addresses in the 169.254.0.0/16 range, which is small enough to lead to collisions in a room with just a few hundred nodes (see the Birthday Paradox). Second, for any IPv4 address outside the link-local range, we cannot be certain that those addresses are not routable on some RAN clients' other network interfaces, raising substantial interoperability issues.

IPv6 link-local addresses have no such problem. There is a canonical and invertible mapping from MAC addresses (that is, 48-bit IEEE 802 addresses) to 128-bit IPv6 link-local addresses [8]. In addition, the IPv6 socket programming interface [11] defines, in the `sockaddr_in6` struct, a field (`sin6_scope_id`) for the purpose of identifying the interface for link-locally addressed connections. This means that client applications can directly address services running on other nodes by MAC address and port. Multicast group addresses at the network layer are also mapped canonically (but non-invertibly) to multicast addresses at the link layer.

Without a distinction between MAC and IP addresses, the IPv6 header is almost entirely redundant for nearly all unicast and multicast traffic. Opportunistic header compression is not a new idea [15], but it works particularly well here: of the 40-octet IPv6 header, we only need to keep one octet in the common case (see §2.6). It is important to note that the compression we use is *opportunistic*: that is, so long as we retain the ability to distinguish between compressed and uncompressed packets, our algorithm is permitted to reject some inputs entirely, forcing them to be sent uncompressed.

2.3 Transport Layer

We briefly consider the performance of TCP on a RAN. Unlike Wi-Fi systems, RANs do not provide reliable delivery at the link layer. Furthermore, for link-local traffic, RAN transport protocols do not need to tolerate packet drops at routers. In this case, the only two sources of loss are stochastic losses at the physical layer, such as collisions or decoding failures, and transmission time-outs.

TCP is designed to interpret lost packets as a sign of congestion in the network and to respond by backing off its congestion window size. However, in the presence of stochastic losses, TCP reacts rather poorly [10]. Many solutions are possible.

One possibility is that if collisions can be eliminated (hypothetically, via perfect carrier sense), then the TCP sender does not need to sense congestion from ACKs at all. This is not to say that the TCP sender will not encounter congestion – fairness is still an issue – but rather that the signaling mechanism of detecting dropped packets should no longer be used to control the congestion window size, and instead the link layer should be polled to determine whether queued packets are reaching the medium or whether the medium is 100% utilized. Only then should the sender back off.

[1] Because our acoustic PHY is approximately 2000 times slower than Wi-Fi, the saved round trip time is on the order of 400 ms.

2.4 Application Layer

Many RAN applications are well served by blind UDP multi-cast; for instance, an indiscriminate file- or link-sharing app might broadcast information to every instance of the same app in the room, or a RAN-enabled "We Have Wi-Fi" sign in a café could broadcast log-in information. These applications require no special system support, except for a canonical mechanism to identify which network interface is the RAN interface. Once the interface is identified, these applications bind a UDP socket to it and call `sendto` or `recvfrom`.

Other applications, like the match-making lobby in a multi-player game, can be divided into two phases: a discovery phase that keeps the UI up-to-date with a list of nearby nodes, and a narrow-cast phase that shares real-time game data. Consequently, fully RAN-equipped devices (as opposed to small, embedded nodes supporting only blind multi-cast) should run an mDNS responder as a system service. Applications access this service via a file-domain socket or other (platform specific) standard inter-process communication mechanism.

An additional service that RAN nodes may provide is connection offloading, which they can advertise via mDNS. The idea is that some RAN physical layers, like BLE, IR, or the acoustic reference PHY, may not be suitable for high-bandwidth or low-latency streaming data. In those cases, the devices may negotiate (over the RAN) a connection over standard Bluetooth, Wi-Fi direct, a mutually-visible access point, WAN unicast via NAT traversal, or, in the worst case, via a mutually-accessible WAN cloud proxy. Fall-back to a cloud proxy would require infrastructural investment beyond what is provided by RAN. Ideally, this case would be highly unlikely due to the prevalence of Wi-Fi and Bluetooth capabilities.

2.5 Application programming interface

A RAN client application sees the RAN in two ways. First, the RAN appears as a standard network interface with only a link-local IPv6 address. The application can bind sockets to this interface as normal, and accept or create connections to other nodes if it knows their addresses. Second, the RAN system service provides the following IPCs:

1. Discover RAN interface ID for use with `sockaddr_in6.sin6_scope_id`
2. Issue mDNS query with timeout
3. Start/stop broadcasting a set of mDNS records

2.6 Header Compression

We compress if, at the sender, we detect a series of headers (beginning from the LLC header) satisfying the following tests:

1. DSAP = `0xaa` (SNAP)
2. SSAP = `0xaa` (SNAP)
3. Control = 3 (Unnumbered information)
4. Protocol ID = 0 (Ethertype)
5. Ethertype = `0x86DD` (IPv6)

---

1. Version = 6 (IPv6)
2. Next Header = 6, 17, or 58 (TCP, UDP, or ICMPv6)
3. Source Address is link-local[2]
4. Destination Address is link-local[2]

We anticipate that these conditions will hold for the vast majority of application traffic.

If any test fails, we transmit a zero octet followed by the unmodified headers. If all tests pass, then we know that the LLC and network headers can be reconstructed from the MAC header. This is because the source and destination MAC addresses are either unicast, in which case the corresponding IPv6 address can be derived via the canonical mapping, or multicast, in which case the low 32 bits of the IPv6 multicast group can be extracted from the canonical mapping. We have already arranged (footnote 2) to only compress in the (common) case where the first 96 bits of any IPv6 multicast addresses are known, so by this reasoning we can completely reconstruct the source and destination IP addresses at the receiver if compression is used.

If the tests pass, we transmit an octet with the value 6, 17, or 58 (based on the Next Header field of the IPv6 header) followed by the TCP, UDP, or ICMPv6 header and payload.

2.7 Reference Physical Layer

Our reference PHY, Blurt, is implemented as an open-source cross-platform library in Python and C++ [17]. For a proven physical layer (PHY) solution, we turned to the 802.11a OFDM scheme with convolutional coding for forward error correction and a 32-bit cyclic redundancy check (CRC) for error detection. This choice of modulation was primarily for expedience; however, it is justified by the high spectral efficiency of OFDM and the long impulse response of the indoor acoustic channel.

We choose a center frequency of 19 kHz, at the upper edge of human hearing (usually given as a ball-park figure of 20 kHz). The response curves of some typical audio hardware are shown in Figure 2. We could have chosen a lower center frequency for more sensitivity, at the cost of a certain shrillness. We use more sub-carriers than Dhwani [22] (64 versus 35) over less bandwidth (2.4 kHz without guard band, versus 6 kHz) so that we tolerate longer multi-path delays. An exemplary Blurt datagram is shown in Figure 3.

[2]If address is multicast, we also require that it matches the prefix `FF02::/96`, which includes mDNS and many other common multicast groups.

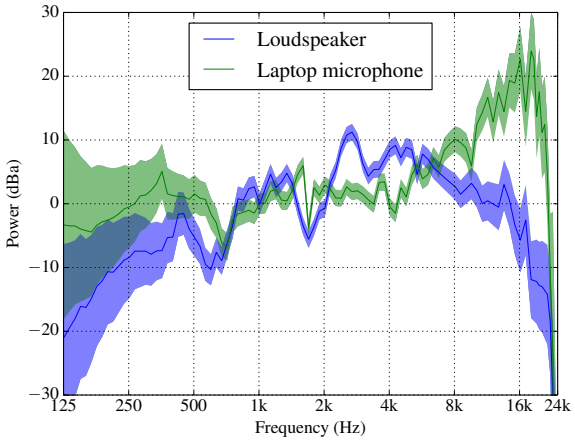


Figure 2: Response curves of a typical desktop loudspeaker and a laptop microphone, as measured against a CM-130 sound pressure level meter.

Range (m)	0.5	1	2	4	8	12
PER	3%	12%	11%	19%	22%	84%

Table 3: Packet error rate (PER) versus range between Blurt senders and receivers (2014 MacBook Pro laptops). Center frequency 19 kHz, bandwidth 3 kHz.

Parameters. The main design parameters in our adaptation of the 802.11a PHY to audio are summarized in Table 4.

Caveats. The acoustic reference PHY throughput ranges from 900 bps in its most robust configuration to 8100 bps in its fastest configuration, with RTTs of 408±4 ms. Offloading traffic onto RF technologies after service discovery through the RAN will be desirable.

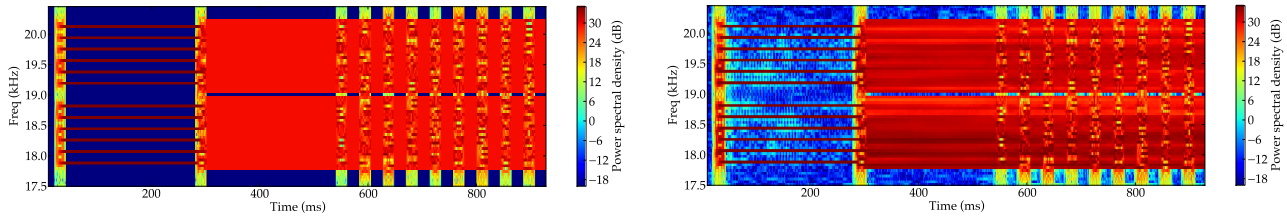
Not every mobile device has an adequate audio passband to support Blurt's inaudible ultrasonic operation. Table 3 shows results from an experiment using MacBook Pro laptops as Blurt senders and receivers. As expected, packet error rates (PER) increase with distance. Blurt successfully delivers packets at ranges up to 12 m.

3. USING THE RAN API

In this section we describe some applications of room-area networks and how they map into our proposed system architecture.

3.1 Guest Access for Wi-Fi Networks

Blurt can be used to establish an authorization scheme where only users in a certain room are able to connect to



(a) Before passing through the channel

(b) After passing through the channel

Figure 3: A Blurt datagram, before and after passing through an acoustic channel. From left to right, the short training sequences (which appear in the frequency domain as 12 broadly-spaced spikes), the long training sequences, and the data symbols.

Parameter	Significance	Value (802.11a/g)	Value (Blurt)
Center frequency	Must be within pass-band of speaker, microphone; should be above limit of human hearing	2.4-5.7 GHz	19 kHz
Bandwidth	Same as above; directly improves throughput	20 MHz	3 kHz
# Sub-carriers	Must be short relative to coherence time of channel but long relative to echo time; impacts PAPR	64	64
Cyclic prefix length	Echo/multi-path tolerance, but cuts into throughput	800 ns	21.3 ms
Training seq. reps.	Determines packet detection and channel estimation accuracy; affects per-packet overhead	2	6
Power	Improves signal at receiver but worsens audibility	< 100 mW	~ 500 mW
Stereo shift	Prevent unintentional beam-forming	0-200 ns	5 ms

Table 4: Design parameters for adapting 802.11a PHY to the acoustic channel.

a given Wi-Fi network, for instance in a company or inside a café, by making the access point (AP) RAN-capable.

Insecure guest access. The user connects to an open Wi-Fi network, and using standard methods is diverted to a captive web portal. The portal challenges the user's device to prove that it is in-room by communicating via the RAN: it includes a tag `<script src="http://auth.BSSID.local/NONCE.js" onerror="retry()" onload="done()">`, causing the user's browser to issue an mDNS query over the RAN. The AP answers this query with an AAAA record pointing to its link-local address. The user's browser then uses this address to open a socket via the RAN, and receives an empty JavaScript file. Now the AP knows that the user is in the room, since the user's browser contacted it via the RAN and provided the correct nonce. The web portal can tell whether the resource was loaded using HTTP header `Access-Control-Allow-Origin` and `<script>` attribute `crossorigin` , and might retry. Note that the user's browser is not trusted to report success. Once the resource request succeeds, the user's browser is redirected to the URL she originally tried to access before reaching the captive portal. All these steps are performed without the user typing in any passwords.

WPA2-encrypted secure guest access. In fact, we can use RAN to provide WPA2-encrypted guest access to an enterprise Wi-Fi network. Today, this would require each guest to be given a Wi-Fi password or personal certificate that then must be entered manually, a somewhat cumbersome task on smartphones.

This time, the AP sends periodic RAN multicast datagrams containing the BSSID and a WPA2 password. A native application running on the user's mobile device listens to this multicast group and displays each advertisement it receives. Once the user selects a network from the list, the application causes the user's device to associate with the corresponding AP. In the process, the device and the AP negotiate a pairwise transient key (PTK) over e.g. EAP-TTLS. The user's device now goes through the in-room challenge detailed above for the case of insecure guest access, with the native application replacing the web browser.

We'd like to give the user confidence that she has not connected to a rogue access point. The in-room challenge nonce is served over TLS with forward secrecy. The server certificate will include the AP's BSSID. The user can now be certain that the TLS connection terminates at a legitimate access point. But a rogue AP could be tunneling the TLS connection to a legitimate AP; we need to bind the TLS connection to the Wi-Fi association. So the AP uses its copy of the PTK to encrypt a known plain-text, and sends the resulting ciphertext to the native application through the TLS connection. Now the application can validate the AP to which it is associated by comparing the BSSID, certificate, and pairwise transient key.

The name-spacing of the TLS certificates bears some discussion. One approach is to have the root certificate signed by a RAN certificate authority (CA), which would have to be trusted. The domain name of the AP would then be of the form BSSID.ran.campus.edu (or company.com). If organizations don't want to reveal their BSSIDs to us, another level of indirection to hide that would suffice.

3.2 Rate Limits and Traffic Prioritization

As we have seen, RAN lets us challenge a device to confirm its location. We can now prioritize traffic from certain rooms (e.g., important meeting rooms), rate-limit traffic from certain rooms, and provide guarantees for traffic from certain rooms. When the user interacts with the captive portal, her location is tabulated with her MAC address on the access point, and used to assign her traffic to a QoS class for routing. To update this information as the user moves, we arrange for her mobile device to advertise an mDNS record containing its Wi-Fi MAC address. Any time the AP would like to update its location mappings, it issues a query over the RAN in each room to learn the set of user devices in that location.

3.3 Building Automation

Recently, Dawson-Haggerty et al. proposed a building operating system called BOSS. RAN provides a useful and interesting way to enforce space-based access control and authorization to control the buildings actuators because it provides evidence of the room in which the device currently is running. Each room of a building could send periodic RAN multicast datagrams containing a cookie that is unique to the room and changes, say, every five minutes. Then the user could supply this cookie to a building application accessible via Wi-Fi to prove their recent location. The same idea can be extended to controlling other devices in a home, campus, or company.

4. RELATED WORK

4.1 Existing Approximations to RAN

Apple's Multi-peer Connectivity Framework [3] enables discovery of and communication with nearby mobile devices using Wi-Fi and Bluetooth. It supports unicast, multicast (up to a limit of 8 peers), and service discovery. However, it does not adhere to room boundaries due to RF leakage through walls. In addition, it uses a proprietary protocol and support is limited to Apple devices. Multi-peer Connectivity optionally uses DTLS to provide security, while our RAN abstraction requires applications to implement their own security.

Sonicnet.js [25] uses audio to enable ultrasonic communication between devices. The library provides applications with a socket-like API. Sonicnet naturally adheres to physical room boundaries, but does not provide service discovery and imposes restrictions on transmitted data due to digital signal processing limitations.

Indoor Localization. Spatial coordinates alone are not sufficient to determine in which room a node is located. It would be possible to augment indoor localization systems [19, 27, 28] with up-to-date floorplan information to build a RAN. One advantage of the acoustic medium is that it does not require floorplan information to respect room boundaries; in fact, mobile nodes can set up a RAN without any support from infrastructure.

4.2 RAN PHY Candidates

Wi-Fi suffers from long association times. This phenomenon has been noted in the opportunistic vehicular communication literature for many years [7], and experience indicates that it has not improved in practice. Hadaller et al. [14] report Wi-Fi setup delays of 13.1 ± 12.3 seconds. Blurt can deliver a large number of packets in this time, and may even be useful for accelerating the Wi-Fi association process.

Bluetooth has further usability issues when secure pairing is required, involving waiting for device discovery and manually copying a PIN code. A number of schemes for improving this situation have been considered previously by Uzun et al. [26]. RANs do not have inherent security guarantees or overhead; these may be added with layering.

Technologies using radio frequency (RF) links are not well-suited for a RAN because it is hard to restrict their communication range to the confines of a physical room. Radio signals usually penetrate walls in offices and living spaces to provide good coverage [23].

Near-field communication (NFC) addresses the pairing problem in a zero-configuration way by requiring close physical proximity (a few centimeters) between nodes. This precludes NFC from being useful for a room-area network.

A RAN using infrared (IR) [13] might seem well-suited because it does not penetrate walls, but IR is highly directional, suffers from dead spots, and may not work in rooms with solar illumination. IR has all but disappeared from mobile devices today.

5. CONCLUSION

In this paper, we have identified a need for room-area networks: a new abstraction in which nodes can hear each other only if they are in the same physical room – that is, within earshot. We have also defined a RAN abstraction and network stack, and we have described example applications from a variety of domains that stand to benefit from this abstraction. Existing technologies such as Wi-Fi, Bluetooth, NFC, and infrared do not meet the needs of RAN. As a proof-of-concept, we have presented Blurt, a portable reference implementation of an 802.11a-like physical layer for the acoustic medium. We anticipate that the availability of RANs on commodity mobile devices will lead to many fruitful developments.

Acknowledgments

We thank the industrial partners of the MIT Center for Wireless Networks and Mobile Computing (Wireless@MIT) for their support. We also thank Madars Virza for thoughtful comments on security properties, Robert Iannucci {Jr.,Sr.} for many helpful conversations, and several anonymous reviewers for constructive feedback.

6. REFERENCES

[1] Android. NFC basics. http://developer.android.com/guide/topics/connectivity/nfc/nfc.html#p2p.

[2] Apple. GKSession class reference. http://developer.apple.com/library/ios/documentation/GameKit/Reference/GKSession_Class/, June 2009.

[3] Apple. Multipeer connectivity framework reference. http://developer.apple.com/library/ios/documentation/MultipeerConnectivity/Reference/MultipeerConnectivityFramework, Sept. 2013.

[4] H. Balakrishnan. Audiocom communication system. http://audiocom602.blogspot.com/, Aug. 2012.

[5] P. Bergel and A. Steed. Data communication system, Apr. 5 2012. US Patent App. 12/926,470.

[6] Building Robotics. Comfy. https://gocomfy.com, Apr. 2015.

[7] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden. A measurement study of vehicular internet access using in situ wi-fi networks. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 50–61. ACM, 2006.

[8] M. Crawford. Transmission of IPv6 Packets over Ethernet Networks. RFC 2464 (Proposed Standard), Dec. 1998. Updated by RFC 6085.

[9] S. Dawson-Haggerty, A. Krioukov, J. Taneja, S. Karandikar, G. Fierro, N. Kitaev, and D. Culler. Boss: building operating system services. In *NSDI*, 2013.

[10] M. Dong, Q. Li, D. Zarchy, P. B. Godfrey, and M. Schapira. PCC: Re-architecting congestion control for consistent high performance. In *NSDI*, 2015.

[11] R. Gilligan, S. Thomson, J. Bound, J. McCann, and W. Stevens. Basic Socket Interface Extensions for IPv6. RFC 3493 (Informational), Feb. 2003.

[12] Google. Guest mode FAQs – Chromecast help. https://support.google.com/chromecast/answer/6109297?hl=en, 2015.

[13] J. Grubor, V. Jungnickel, and K.-D. Langer. Adaptive-modulation technique in wireless infrared indoor communications. In *Photonic Networks, 2006 ITG Symposium on*, pages 1–8, April 2006.

[14] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal. Vehicular opportunistic communication under the microscope. In *Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 206–219. ACM, 2007.

[15] J. Hui and P. Thubert. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. RFC 6282 (Proposed Standard), Sept. 2011.

[16] A. Huibers. Bump validation, Nov. 5 2013. US Patent 8,577,292.

[17] P. Iannucci. Blurt codebase. http://www.github.com/piannucci/blurt.

[18] A. Kauffmann and B. Smus. Tone: An experimental Chrome extension for instant sharing over audio. http://googleresearch.blogspot.com/2015/05/tone-experimental-chrome-extension-for.html, May 2015.

[19] S. Kumar, S. Gil, D. Katabi, and D. Rus. Accurate Indoor Localization with Zero Start-up Cost. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 483–494, New York, NY, USA, 2014. ACM.

[20] H. Lashkari, H. Le, and D. Borges. Peer-to-peer file transfer between computer systems and storage devices, Sept. 16 2014. US Patent 8,838,697.

[21] K. Lea. Wham City Lights. http://whamcitylights.com/, Mar. 2014.

[22] R. Nandakumar, K. K. Chintalapudi, V. Padmanabhan, and R. Venkatesan. Dhwani: Secure peer-to-peer acoustic nfc. *SIGCOMM Comput. Commun. Rev.*, 43(4):63–74, Aug. 2013.

[23] T. S. Rappaport and S. Sandhu. Radio wave propagation for emerging wireless personal communication systems. In *Wireless Personal Communications*, pages 1–27. Springer, 1995.

[24] C. Roeding and A. Emigh. Method and system for presence detection, 02 2011.

[25] B. Smus. Ultrasonic networking on the web. http://smus.com/ultrasonic-networking/, Aug. 2013.

[26] E. Uzun, N. Saxena, and A. Kumar. Pairing devices for social interactions: a comparative usability evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2315–2324. ACM, 2011.

[27] J. Xiong, K. Jamieson, and K. Sundaresan. Synchronicity: Pushing the Envelope of Fine-grained Localization with Distributed Mimo. In *Proceedings of the 1st ACM Workshop on Hot Topics in Wireless*, HotWireless '14, pages 43–48, New York, NY, USA, 2014. ACM.

[28] J. Xiong, K. Sundaresan, and K. Jamieson. ToneTrack: Leveraging Frequency-Agile Radios for Time-Based Indoor Wireless Localization. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, pages 537–549, New York, NY, USA, 2015. ACM.