High-Confidence Computing 4 (2024) 100223

Contents lists available at ScienceDirect

High-Confidence Computing

journal homepage: www.sciencedirect.com/journal/high-confidence-computing

Research article

Aquilo: Temperature-aware scheduler for millimeter-wave devices and networks

Moh Sabbir Saadat *, Sanjib Sur, Srihari Nelakuditi

Department of Computer Science and Engineering, University of South Carolina, SC 29201, USA

ARTICLE INFO

Article history:
Received 7 December 2023
Revised 5 January 2024
Accepted 19 February 2024
Available online 27 March 2024

Keywords:
5G
Millimeter-wave
Temperature-aware design
Multi-antenna scheduler

ABSTRACT

Millimeter-wave is the core technology to enable multi-Gbps throughput and ultra-low latency connectivity. But the devices need to operate at very high frequency and ultra-wide bandwidth: They consume more energy, dissipate more power, and subsequently heat up faster. Device overheating is a common concern of many users, and millimeter-wave would exacerbate the problem. In this work, we first thermally characterize millimeter-wave devices. Our measurements reveal that after only 10 s of data transfer at 1.9 Gbps bit-rate, the millimeter-wave antenna temperature reaches 68°C; it reduces the link throughput by 21%, increases the standard deviation of throughput by 6×, and takes 130 s to dissipate the heat completely. Besides degrading the user experience, exposure to high device temperature also creates discomfort. Based on the measurement insights, we propose *Aquilo*, a temperature-aware, multi-antenna network scheduler. It maintains relatively high throughput performance but cools down the devices substantially. Our testbed experiments under both static and mobile conditions demonstrate that *Aquilo* achieves a median peak temperature only 0.5°C to 2°C above the optimal while sacrificing less than 10% of throughput.

1. Introduction

The explosive demand for mobile broadband globally has created significant stress on the existing wireless infrastructure. Millimeter-wave (mmWave) has emerged as the core, new technology for the next-generation wireless LAN and cellular standards: IEEE 802.11ay [1] and 5G NR [2]. MmWave systems are the key enabler for applications that demand multiple Gbps throughput and ultra-low latency connectivity — virtual and extended reality, tactile internet, telesurgery, control for smart infrastructures, and autonomous vehicle safety [2–4]. MmWave systems achieve these capabilities by operating at a very high frequency and ultra-wide bandwidth, on the order of multiple GHz.

Such a high operational regime, however, brings unique challenges: Compared to the micro-wave devices, like Wi-Fi and LTE, the mmWave devices consume more energy, dissipate more power, and subsequently heat up faster. Increase in device temperature not only affects the hardware but also is disconcerting to the users, especially when devices are small [5], hand-held [4,6,7], body-worn [8,9], and near to the face and brain [10–13]. For example, Fig. 1(a) shows that the surface of a running mmWave device heats up very quickly. Device overheating is a common concern of many users, and mmWave would exacerbate the problem further. Thus, investigating ways to mitigate thermal-inefficiencies in mmWave devices is of vital importance.

Existing research works have extensively characterized mmWave channel, link, network, and applications; however, the thermal characteristics of the mmWave device are relatively less understood. Compared to the micro-wave devices, mmWave devices operate at a very high frequency and ultra-wide bandwidth. Thus, each hardware component, *e.g.*, baseband, ADC/DACs, PLLs, mixers, power amplifiers, *etc.*, consumes more energy and dissipates more power. Although mmWave devices are more energy-efficient (consume less energy per bit) than Wi-Fi or LTE [14,15], higher aggregate energy consumption and power dissipation can heat the mmWave devices quicker. While existing works have looked into the power consumption of the commercial mmWave devices [14,15], to the best of our knowledge, *none has looked at the thermal characteristics of the mmWave devices.*

To this end, we characterize the thermal profile of a Commercial-Of-The-Shelf (COTS) 60 GHz mmWave smartphone [7] under various conditions. Our measurements reveal that after only 10 s of data transfer, at room temperature, with 1.9 Gbps bit-rate, the mmWave antenna temperature reaches up to 68°C. It reduces the average link throughput by more than 21%, increases the standard deviation of throughput by 6×, and takes about 130 s to dissipate the heat. Such excessive temperatures of 5G NR devices are also reported by consumers of different devices and brands [16,17] as well as by recent publications [18,19]. Besides

* Corresponding author.
E-mail address: msaadat@email.sc.edu (M.S. Saadat).

https://doi.org/10.1016/j.hcc.2024.100223

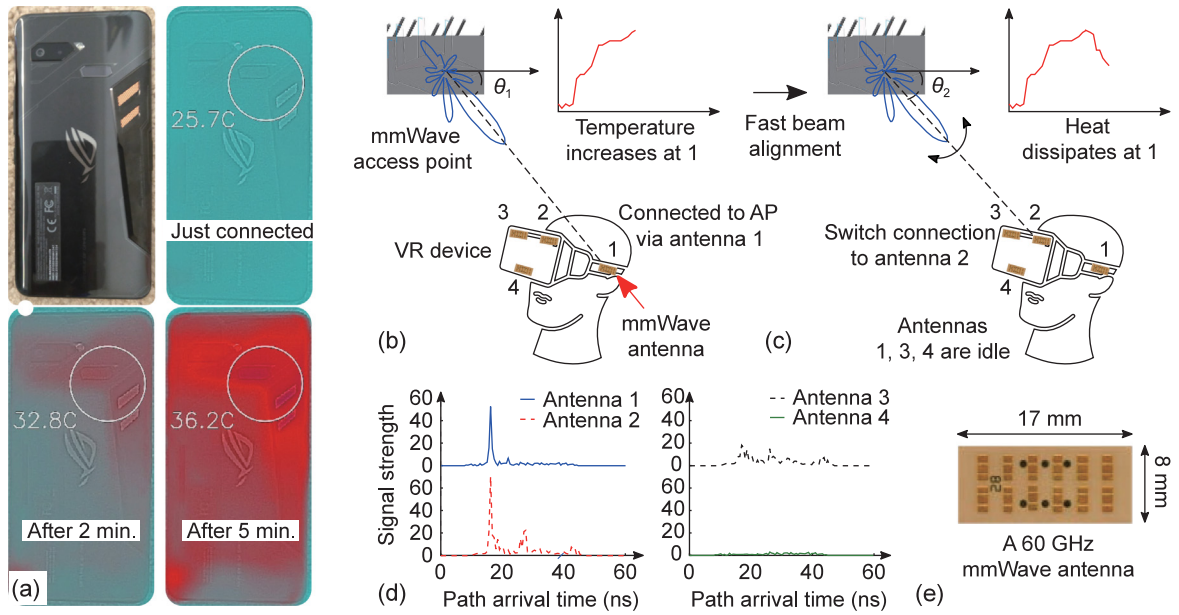

Fig. 1. (a) Thermal image of a networked millimeter-wave (mmWave) smartphone [7]. The core idea behind *Aquilo*: (b) VR device with 4 mmWave antennas connected to the access point (AP) via antenna 1, and data transfer increases its temperature; (c) Scheduler switches to relatively cooler antenna 2 with the best link to the AP; (d) Signal strength of the best beam from the 4 antennas; (e) A 60 GHz mmWave antenna [6].

degrading the user experience, an increase in device temperature is disconcerting. More importantly, exposure to a high temperature may not only create short-term discomfort, but also could have long-term adverse health effects [20,21].

Driven by the measurement insights, we propose *Aquilo*[1] — a temperature-aware multi-antenna scheduler that cools down mmWave devices substantially. *Aquilo*'s key idea is intuitive: Before one antenna heats up excessively, its data stream may be switched or distributed to other redundant antennas, allowing it to dissipate the heat (see Fig. 1[b-c]). Equipping a mmWave system, like 5G smartphone or VR, with multiple antennas is not only a reasonable system's choice, since the mmWave antennas are tiny (Fig. 1[e]), but also is necessary to provide reliable connectivity under channel fluctuations and obstructions [6,22,23]. Coordination among these multiple antennas *proactively* can reduce the overall temperature.

This idea draws inspiration from the existing *thermal mitigation* techniques in multi-processor architectures [24–28]; yet, the challenges are in the variable thermal behavior and variable connectivity of the mmWave antennas (Fig. 1[d]). We propose an adaptive multi-antenna scheduler that exploits the recent observation of the thermal profiles, and probe and switch scheme to maintain relatively stable throughput and reduce the device temperature.

While we motivate *Aquilo* with chip-level thermal characterization in [18], in this work, we further show that mmWave devices may suffer from overheating at the external surface as well. Moreover, *Aquilo* is originally designed and proposed for a four-antenna device in [18], which may be a suitable choice for smartphones. However, not every device can support four antennas (*e.g.* smartwatches may support only two antennas) whereas some devices can support more than four antennas (*e.g.* a VR headset). This largely depends on the form factor of a mmWave antenna as illustrated in Fig. 1(e). To this end, we scale *Aquilo* to minimize temperature with variable number of antennas. A key challenge in temperature-aware scheduling with an increased number of antennas is the higher computational cost due to the larger search space. To counter this, we propose an adaptive Aquilo that further reduces the search space to achieve similar temperature performance as the non-adaptive one while reducing the computational complexity significantly. Furthermore, we consider *Aquilo* under a trigger-temperature mode, *i.e.* the system will run without temperature-aware scheduling until a pre-set threshold fpr the system temperature is reached at which point *Aquilo* will be triggered. With these enhancements, Aquilo aims to not only scale with a different number of antennas on mobile devices but also strike a good balance between minimizing temperature and maximizing throughput under practical settings.

We validate *Aquilo* on a 60 GHz mobile mmWave testbed; it consists of a NETGEAR X10 Access Point (AP) [29] and ASUS ROG smartphones [7]. Both the AP and smartphones are IEEE 802.11ad standard-compliant; also, ASUS ROG is one of the commercially available 60 GHz smartphone, currently. Since the COTS smartphone is equipped with one mmWave antenna only, we collect real device throughput and temperature profiles, but use trace-based methods to emulate a multi-antenna device. Our testbed experiments demonstrate that, in comparison to a *throughput-only maximization* scheduling, *Aquilo* can effectively reduce the median peak temperature by 12°C and 9.5°C, under static and mobile conditions, respectively. While these improvements come from sacrificing 9.8% (static) and 8.5% (mobile) of throughput, *Aquilo* is still able to support at least 1.4 Gbps throughput at all times in static and 1.03 Gbps in more than half of the mobile cases. Furthermore, we show that scaling the number of antenna from 2 to 8 improves the temperature minimization by 11°C in spite of the restriction imposed by the updated *Non-Adjacency Criteria*. Our field trials with traces collected from two applications, FTP and VR, show that *Aquilo* satisfies the minimum throughput requirements while simultaneously achieving the near-optimal device temperatures. We also show that *Aquilo* can be triggered at higher threshold temperatures and still achieve fair temperature minimization.

In summary, we have two contributions: (1) ***Thermal Characterization of 60 GHz mmWave Device*** To the best of our knowledge, we are the first to perform a systematic study of the effects of mmWave device states and link performance on the device's

[1] *Aquilo* was the Roman god of cold north wind and bringer of winter.

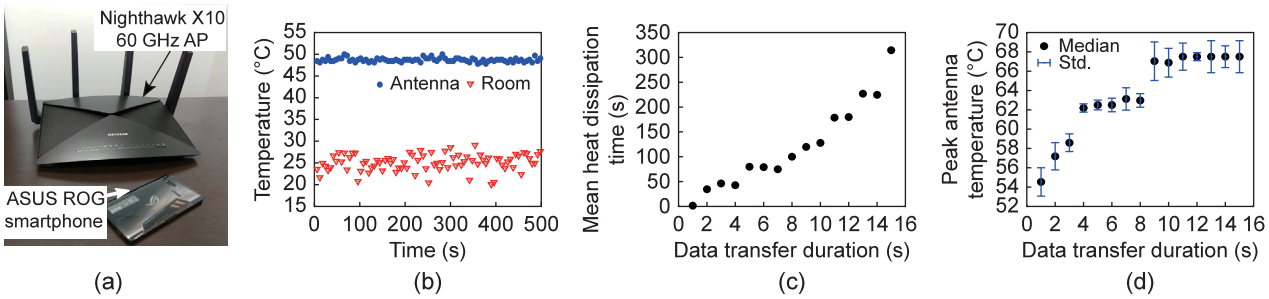

Fig. 2. (a) Millimeter-wave (mmWave) AP and smartphone. (b) The temperature of the room and the mmWave antenna at idle state. Effect of data transfer on: (c) Mean heat dissipation time; and (d) Peak temperature.

temperature and the effect of temperature on the device's performance [18]. (2) ***Temperature-Aware, Scalable Multi-Antenna Scheduler*** Based on the insights from our thermal characterization, we propose, design, and validate a temperature-aware multi-antenna scheduler and demonstrate its effectiveness and scalability in maintaining the link performance while reducing the temperature substantially. We have open-sourced our thermal characterization dataset and multi-antenna scheduler code in [30].

2. Background and related work

Millimeter-Wave Devices and Standards MmWave devices operate at a very high frequency and ultra-wide bandwidth. Currently, there are two most popular mmWave standards: 5G NR [2] (frequency ranges are 26.5–29.5 GHz and 37.0–40.0 GHz); and IEEE 802.11ad [31] (57–71 GHz). Specifically, IEEE 802.11ay devices operate on the unlicensed 60 GHz mmWave, use 2.16 GHz bandwidth, and can achieve peak bit-rate up to 7 Gbps. Since mmWave channel suffers from high signal propagation loss, both the standards use phased-array antenna and directional beam for signal strength compensation. Due to the small form factor of mmWave radio-frequency (RF) components and elements, multiple antennas can be integrated into mobile devices [6,22,23]. Besides, multiple antennas provide reliable connectivity under channel fluctuations and obstructions [32–36]. While existing research works have extensively characterized mmWave channel [37–44], link [45–50], network [51–58], and applications [59–64], *Aquilo* is the first to characterize the thermal profile of mmWave antennas, and design, evaluate, and demonstrate a temperature-aware, scalable multi-antenna scheduling scheme. *Aquilo* is complementary to the existing mmWave systems' research and can benefit from their reliable connectivity and improved performance.

Thermal Mitigation Techniques *Aquilo* is partly inspired by the thermal mitigation techniques in multi-CPU systems. Several proposed approaches can reduce the temperature significantly: dynamic trigger [24,65,66]; hybridized thermal stress-aware adaptation [25]; priority queueing [26]; passive load balancing and active migration [27]; and stochastic techniques [28]. Yet, *Aquilo* faces two challenges that are absent in multi-CPU systems: variable thermal behavior; and highly variable and unpredictable connectivity of the mmWave antennas. While there are existing works in micro-wave systems to minimize the power or energy consumption [67–72], *Aquilo* is the first work to look into minimizing the system-level temperature of active mmWave antennas.

3. Thermal characterization

In this section, we characterize the impact of device's states and throughput performance on its thermal profiles. *First*, we investigate the thermal behavior of a mmWave device: The device's surface temperature as well as the antenna temperature during the active and idle states. *Second*, we identify performance fluctuations under high antenna temperatures. *Finally*, we study the trade-off between throughput performance and antenna temperature by inducing bit-rate adjustment and periodic idleness. In all cases, we explain the underlying reason for the observed behavior and how it differs from what may be expected. The findings of this section inform our proposal for temperature-aware multi-antenna scheduler.

Setup For our experiments, we use commercial mmWave smartphones: ASUS ROG [7] and AP: NETGEAR X10 [29] (Fig. 2[a]). The devices operate at 60 GHz mmWave and are IEEE 802.11ad compliant [31]. They use Qualcomm mmWave chipset [73] and operate on a 2.16 GHz bandwidth, support up to 4.62 Gbps bit-rate, and have a 32-elements phased-array antenna [51,53]. Furthermore, the devices have embedded, high-resolution temperature sensors that allow us to monitor the antenna's temperature continuously. The smartphones use Android version 8.1 with a Linux kernel, and we can access the temperature data directly from the kernel file: `/sys/kernel/debug/ieee80211/phy0/wil6210/temp`. We performed experiments in a temperature-controlled indoor office environment with static mmWave channel conditions.

3.1. Thermal profile in idle and active states

Profile in the Idle State We start by understanding the impact of the device's idle and active states on the thermal profile. We first measure the temperature when the mmWave antenna is idle, *i.e.*, no data communication with the AP. Fig. 1(a) shows that in the idle state, the temperature of the device surface is at 25.70°C, measured with a FLIR One thermal camera [74]. This is close to the ambient temperature of the measurement area. Fig. 2(b) shows that the antenna's average temperature is 48.63°C, about 24°C higher than the room temperature. This high idle temperature is due to the very high power consumption during idle listening [15,71] — the smartphone needs to listen to the incoming mmWave packets and assess the clear channel condition continuously [31]. This also corroborates with prior measurement studies that show that the idle listening power consumption in mmWave devices can be between 1–1.7 W [15,75].

Profile in the Active States

Active states consume more power (∼2.5 W) during data transfer [15,71]. So, it increases the antenna temperature further. Nonetheless, we expect that, as soon as the data transfer stops, the antenna would cool down and reach its idle temperature quickly. But heat acts just like stored energy in a capacitor — it takes a long time to dissipate, and more the data transfer, longer the heat dissipation time. To understand this effect, we set up the smartphone 1 m away from the AP and transfer data at a peak rate for a specific duration; the duration varies between 1 s and 15 s., and we measure the antenna temperatures.

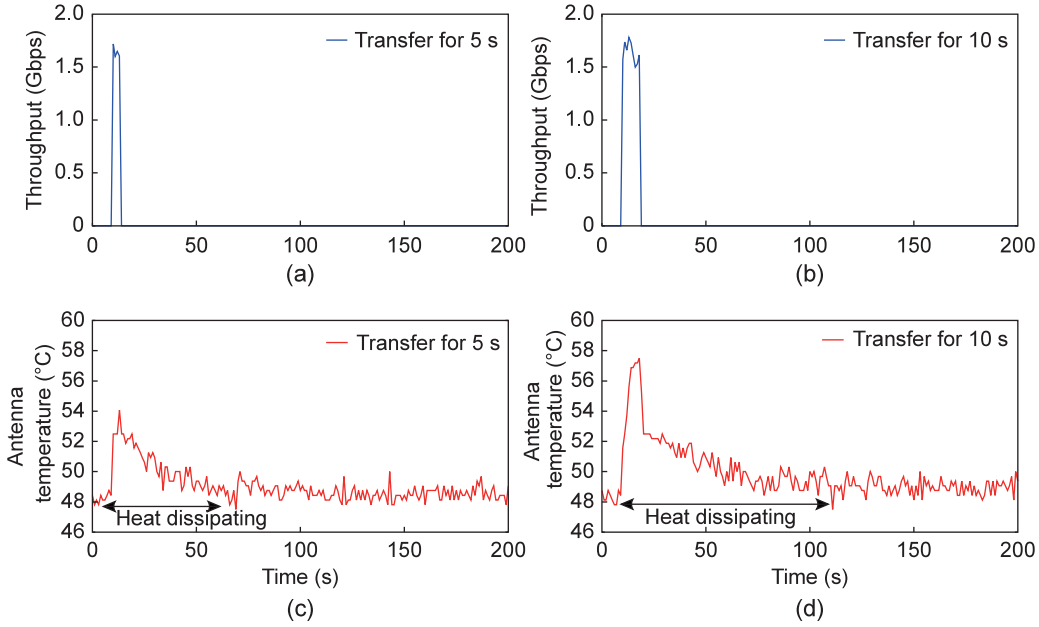

Fig. 3. (a–b) Data transfers for 5 and 10 s. (c–d) Antenna temperatures vs. time and heat dissipations.

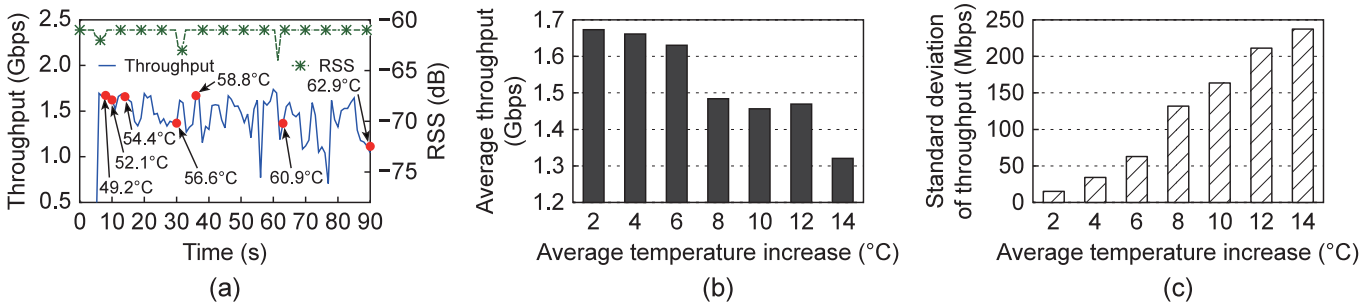

Fig. 4. Link performance is affected by higher antenna temperature: (a) RSS is stable, but throughput fluctuates over time; every time index is marked for 2°C temperature increase; For every 2°C increase, we see changes in (b) Average throughput; and (c) Standard deviation of throughput.

Figs. 3(a–d) plot two examples of antenna temperature for 5 s and 10 s of data transfers, where the 10 s of transfer requires about 130 s for heat dissipation. The dissipation time is measured as the time it takes for the antenna to return to its idle temperature. Increase in the transfer duration increases the dissipation time as shown in Fig. 2(c). More importantly, *the average dissipation time can be up to 20× higher than the transfer duration*. The transfer durations in our experiments are lower than a typical application run-time; but certain applications, such as wireless AR/VR streaming, require continuous Gbps data transfer. Besides, higher data transfer duration affects the antenna's peak temperature too. Fig. 2(d) shows that the temperature can reach up to 68°C for only a 10 s. data transfer.

To study how quickly heat from the antenna percolates the device's surface, we measure the smartphone's surface temperature with the FLIR One thermal camera from the point of boot-up and during the device's active state. Fig. 1(a) shows that the device's surface temperature increases from the room temperature 25.70°C to 36.20°C, *i.e.*, an increase of ≈ 11°C, within only five minutes. Fig. 1(a) further shows that even though the heat spreads throughout the device surface, the antenna region continuously generates more heat. Such increased surface temperature could be a source of discomfort to the user, degrading not only the experience and but also potentially causing long-term adverse health effects.

3.2. Performance fluctuations with high temperature

High antenna temperature can adversely affect the throughput performance too, due to increased thermal noise and leakage current. To quantify this effect, we perform experiments under the previous setup, but change the AP-smartphone distance to 50 cm. *First*, we transfer data at a peak data rate for 90 s. duration continuously and measure the antenna temperature as it rises up. *Then*, we divide the measurement into several time segments, marking every time index where the temperature increases by approximately 2°C. *Finally*, we measure the average and standard deviation of the throughput for each time segment. We repeat our experiments 85 times and present average of the results. Throughout the experiments, we also measure the Received Signal Strength (RSS) by the AP. The standard deviation of the RSS is 0.47 dB only; in other words, the data is transferred at a very stable channel condition.

Fig. 4(a) shows an example RSS and throughput profile and marks the time indices for each segment. Even under the stable channel condition, the throughput fluctuates significantly. Figs. 4(b–c) plot the average throughput and its standard deviation for every 2°C rise in temperature, up to 14°C maximum increase. Clearly, average throughput degrades as the temperature increases; the link loses 21% of average throughput when temperature increases by 14°C. Besides, the standard deviation of the throughput increases by almost 6×; it can reach up to

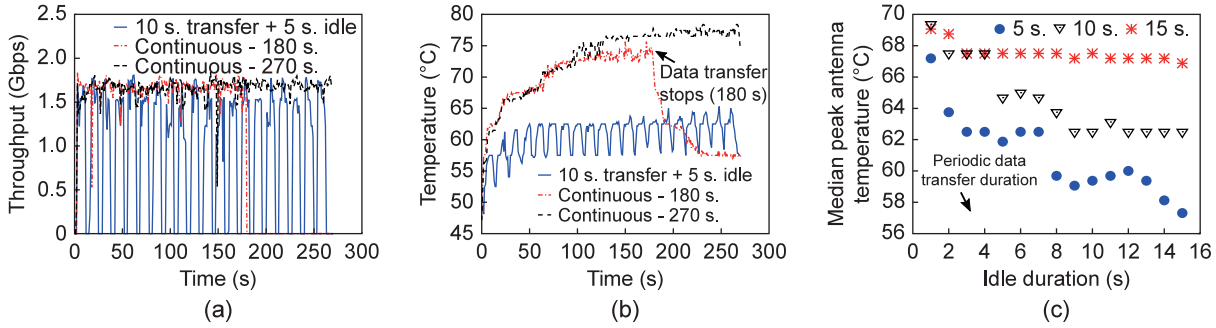

Fig. 5. Under three data transfer modes: 10 s. transfer and 5 s. idle, Continuous −180 s., and Continuous −270 s. (a) Throughput of the millimeter-wave link; and (b) Temperature profiles of the antenna. (c) Effect of various idle durations on antenna's peak temperature.

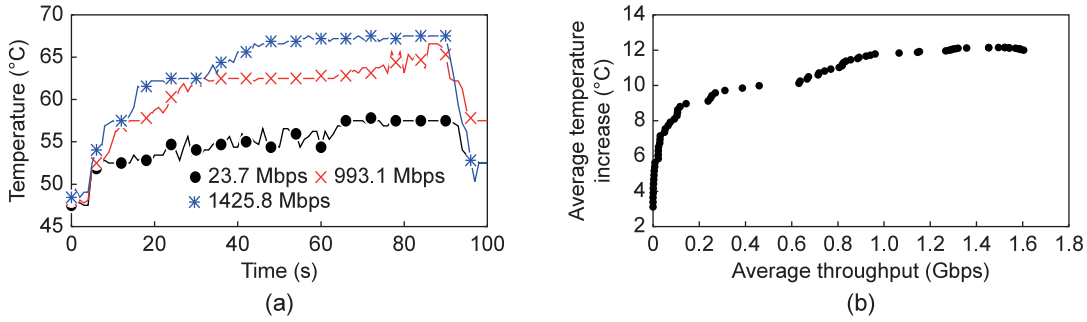

Fig. 6. (a) Temperature profile of 85 s of data transfer; (b) Average temperature increase for different throughputs.

240 Mbps. Thus, *higher antenna temperature not only degrades average performance, but also increases its variations significantly*. Since both noise and leakage current degrades signal fidelity, we speculate that the device adapts to it by changing the bit-rate. But commodity devices currently do not allow measuring such fine-grained bit-rate adaptation, thermal noise, or leakage current; so, we leave an extensive analysis of the performance fluctuations under high temperature as future work.

3.3. Effect of periodic idleness and bit-rate adjustment

Inducing Periodic Idleness A strawman approach to reducing the antenna temperature is to keep it idle in between the data transfers, since *an idle period helps the antenna to cool down*. To understand this effect, we use the previous experimental setup and run experiments for 270 s with three transfer modes: continuous for 270 s; periodic for 10 s followed by idleness for 5 s (*i.e.*, 66.7% duty cycle); and continuous for 180 s (66.7% of 270 s). Figs. 5(a–b) show the throughputs and temperature profiles. For continuous 180 s and 270 s transfers, the temperatures grow steadily, peaking at 75.63°C and 79.38°C, respectively. After 180 s, when the transfer stops, the antenna starts cooling down; but it never reaches the idle temperature within 270 s. Introducing a periodic idleness of 5 s after every 10 s of transfer reduces the temperature substantially; the peak temperature is below 64°C, for 96% of the time. Fig. 5(c) also shows that longer idleness in between transfers can reduce the peak temperature when the transfer durations are short. But this is ineffective for a longer transfer duration, *e.g.*, when the data transfer duration is periodic 15 s, none of the idle durations from 3 to 15 s. can reduce the peak temperature below 66°C.

Inducing Bit-Rate Adjustment Another approach to reducing the temperature would be to slow down data transfer speed. *The device consumes less power to transfer at a lower bit-rate, thus, heats up slowly*. This is reflected in Fig. 6(a); it illustrates three experiments at different mean throughput-level. For the mean throughputs at ∼1.4 Gbps, ∼0.9 Gbps, and ∼0.024 Gbps, the peak temperatures are at 67.5°C, 63°C, and 56°C, respectively. However, Fig. 6(b) shows that the temperature cannot be reduced by changing the throughput-level anywhere from 1.6 to 1 Gbps, hardly 0.17°C. Furthermore, for certain devices and applications, *e.g.*, wireless VR/AR, *neither increased idle duration nor reduced bit-rate is affordable since they require stringent throughput and latency guarantees*; but it may be possible to switch the data stream to another antenna with a lower temperature.

3.4. Measurements summary

In summary, we showed the following properties:

(1) MmWave device's temperature could be high, even when the device is idle; moreover, a longer data transfer duration not only increases the peak temperature, but also takes significantly longer time for heat dissipation — the dissipation time can be up to 20× higher than the transfer duration.

(2) High device temperature, in turn, affects the link performance: even at static conditions, it reduces the average link throughput by more than 21% and increases the standard deviation of throughput by 6×.

(3) A lower bit-rate can reduce the device temperature, only when the link throughput is below 1 Gbps; more importantly, periodic idleness can help reduce antenna's temperature, but only when the data transfer duration is short.

4. Aquilo design

Driven by the measurement insights, we propose *Aquilo*, a thermal profile based multi-antenna scheduler to maximize the link performance and minimize the antenna temperature. *Aquilo* enables a mmWave AP and end-user device to select relatively cooler antennas dynamically. While existing works in microwave systems reduce the idle temperature by scheduling sleep periods [31,67–70] or reducing ADC bit-rate or clock-rate [71,72],

to the best of our knowledge, *none has looked at minimizing the system-level temperature of active mmWave antennas.*

Our key idea stems from the observation that the idle period between data transfers helps the cooling process. Unfortunately, periodic idle durations in a single antenna system are unaffordable for many applications, *e.g.*, wireless VR/AR, real-time streaming, because they require stringent throughput and latency guarantees. Fortunately, upcoming mmWave systems, like 5G NR smartphones and VR/AR devices, are being equipped with multiple antennas. It is not only a reasonable system's choice, but also necessary to provide reliable connectivity [6,22,23]. *Aquilo* leverages the presence of and coordination among these multiple antennas to reduce the overall system's temperature. However, the objectives of high throughput and low device temperatures are perennially in conflict. So, a key networking challenge we aim to solve is providing an uncompromised quality of throughput and latency experience, while simultaneously maintaining a near-optimal system's temperature. To this end, we design a multi-antenna, online thermal-profile driven, look-ahead network scheduler. We now describe each component in detail.

Multi-Antenna Coordination At a high level, before one antenna heats up excessively, its data stream may be switched to the other antennas, allowing it to dissipate heat. This idea is partly inspired by the *thermal mitigation* techniques in multi-CPU systems [24–28,65,66]. But there exist significant differences between CPU's and mmWave antenna's working principle. *First*, while any one of the CPUs can be turned on and expected to work, such an assumption is invalid for the mmWave antennas; this is because channel fluctuations and antenna obstructions may not allow for a link establishment that is strong enough for the required throughput of the running application. *Second*, the thermal behavior of the antenna quickly changes depending on the link performance and user's handling of the device, in addition to device insulation and the surrounding temperature. So, multi-antenna coordination techniques will need to consider at least two mmWave issues not present in the current thermal mitigation techniques: (1) *variable thermal behavior*; and (2) *highly variable and unpredictable connectivity*.

To address these two issues, *we propose using an online thermal profile estimation based look-ahead scheduling among multiple mmWave antennas*. Data transfer from an antenna increases its temperature while idleness decreases it. Furthermore, the increase and decrease rates depend on the amount and duration of transferred data (Section 3). Thus, if we could somehow predict an antenna's temperature when they are transferring data at a future point in time, we would be able to schedule a set of antennas such that the peak temperature is minimized and certain performance criterion is met. Unfortunately, the temperature increase and decrease rates are not always deterministic; besides, it is hard to predict mmWave link performance ahead of time because of highly variable connectivity. *Aquilo* leverages near-past observations of the thermal profile for the temperature prediction and a look-ahead schedule and probe scheme for antenna selection. Next, we describe these design components in detail.

4.1. Thermal profile estimation

Issues with a Fixed Thermal Profile Model A natural way to predict an antenna's future temperature would be to use a fixed temperature rise and fall model. Since the model parameters depend on the data transfer rate and duration, based on several measurements, we could extract the parameters and use them during the run-time. Unfortunately, the model itself varies depending on several factors: device types; insulation materials, locations, and amount; surrounding temperature; and user's handling of the device. To understand the model variations, we measure the thermal profile using the AP-smartphone set up under two environments: indoor office; and home. We set up the smartphone 50 cm. away from the AP and transfer data at a peak rate for 10 s. continuously, ensuring that the channel is stable and the average throughput is around 1.6 Gbps under both the environments. Figs. 7(a–b) show the resulting thermal profiles under the two environments. Clearly, they differ significantly in terms of three parameters: peak temperature; heat dissipation time; and noise. Thus, a fixed temperature rise and fall model established from pre-measured dataset will not be generalizable.

Online Profile Estimation Instead of relying on a fixed model, *Aquilo* leverages online measurements from near-past time, by recording the temperature of the antennas when they are active or idle. But from our observation in Figs. 3 (Section 3) and Figs. 7(a–b), clearly, we need to model the active and idle states separately; this is because switching an antenna from active to idle state stops increasing and starts decreasing the temperature immediately, creating a stark discontinuity in the thermal profile. Thus, the thermal profile of a mmWave antenna can be modeled as two exponentials: *exponential gain*, $e^{\alpha \cdot t}$, in the active state; and *exponential fall*, $e^{-\beta \cdot t}$, in the idle state. Furthermore, we can estimate the α and β parameters from the near-past temperature observations. Despite this discontinuity, these separate models serve us better in estimating the thermal profiles accurately; the average root mean square error for model fit across all our measured thermal profiles is only 0.52°C. Figs. 7(a–b) also show two such fitting results over the raw measurements. Unfortunately, such raw data is unavailable when the device boots up, or the antenna wakes up for the first time. In such cases, *Aquilo* can randomly select and switch between antennas to bootstrap the thermal profiles. Note that, such random selection and switch happen only once per antenna since when an antenna starts data transfer, we can measure its thermal profile immediately.

4.2. Look-ahead schedule, antenna probe and switch

Peak Temperature Based Schedule *Aquilo* leverages the estimated thermal profiles to find a list of candidate antenna schedules. The profile indicates the likelihood and change of an antenna's temperature when it is assigned for data transfer at a future point in time. So, under each schedule, the antennas will go through a temperature transformation and would reach a different peak temperature potentially. Therefore, we can pre-compute the peak temperature attained by each schedule and select the one with the lowest peak. Selecting a schedule with lowest peak temperature also ensures that there is no possibility of a significant temperature differential, *i.e.*, one antenna heating up too high while the others are remaining at idle states. Let us consider a simple example to illustrate this point. Assume we have a 3 antenna system; each antenna is initially at idle temperature, 48°C. We would like to schedule antennas every 1 s. interval; therefore, for a total of 3 s., there are 27 (3^3) possible schedules: (1, 1, 1); (1, 1, 2); . . .; (1, 2, 3); . . .; (3, 3, 3). Furthermore, let us assume that each antenna's temperature increases by 2°C every 1 s. when they are scheduled and transfer data. Hence, we can pre-compute the peak temperature of the schedules: (1, 1, 1) → 54°C; (1, 1, 2) → 52°C; . . .; (1, 2, 3) → 50°C; . . .; (3, 3, 3) → 54°C. Clearly, schedule (1, 2, 3) has the lowest peak temperature, and choosing it for future data transfer will minimize the system temperature.

Unpredictable Connectivity The above simple example assumes that there is an equal likelihood of selecting any one of the three antennas. In mmWave, however, the connectivity is highly variable and unpredictable: channel fluctuations and rampant obstructions from the user's hand, body, and various environmental objects may block one or many antennas [41,44–48,51,76]. Thus, the antennas may not have a strong enough link

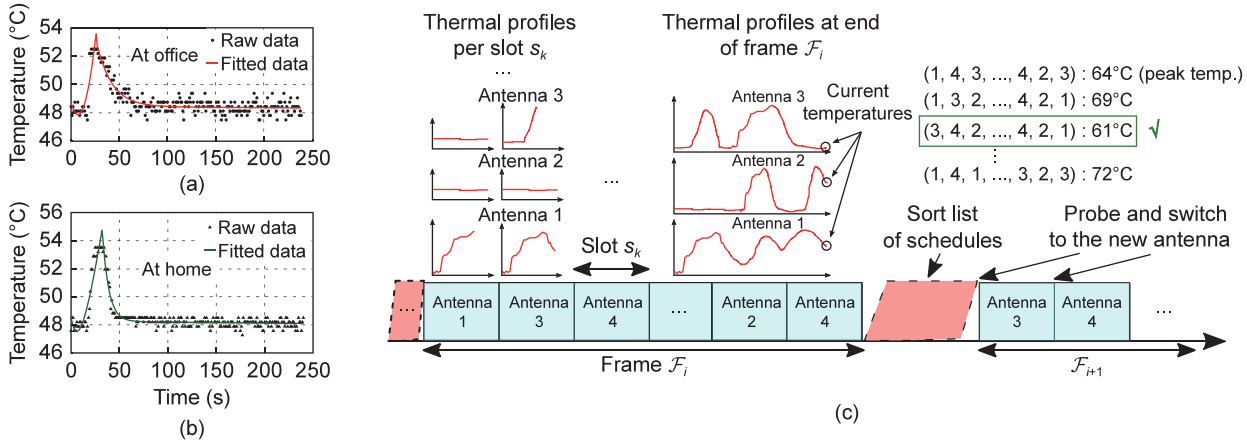

Fig. 7. Data transfer for 10 s., and then, heat dissipation: (a) At office; and (b) At home. (c) *Aquilo* schedules antennas at the beginning of every frame $\mathcal{F}_i$; then, probes and switches to a new antenna every slot s_k.

or sustain an application's performance requirement. To overcome this challenge, *Aquilo* proposes an antenna probing scheme before selecting and switching to it. *First*, *Aquilo* sorts the list of potential schedules as per the likelihood of the lowest peak temperature and then selects the first choice; but there is no guarantee on the first antenna's performance in the selection. *Next*, *Aquilo* invokes a fast beam alignment protocol [31,48,53] to probe for link strength towards the AP, and using an effective SNR metric [77] converts the strength to throughput performance. *Finally*, *Aquilo* switches to the new mmWave antenna that satisfies the expected performance requirement of the application.

Since the first selection may not have sufficient link strength, *Aquilo* iterates through the list of sorted schedules to eliminate the ones beginning with the antenna without the link. It, then, selects the first antenna from the resulting list, which ensures a good quality link as well as a lower peak temperature. Still, there could be scenarios where no antennas in the selected schedule have any link towards the AP. In such cases, *Aquilo* falls back to the currently active antenna to at least guarantee connectivity. Furthermore, antenna probing and switching take a relatively small amount of time with state-of-the-art beam alignment protocols (less than 0.5 ms. [48,53]); thus, the latency overhead from *Aquilo* will be very low.

Practical Considerations Ideally, *Aquilo* should estimate the thermal profiles, sort the list of schedules, and probe and switch to a new mmWave antenna continuously. However, such continuous operations will not only be computationally expensive, but also incur high probing and switching overheads. For practical considerations, *Aquilo*, thus, operates in a chunk of discrete-time, which we call a *frame* $\mathcal{F}$. At the beginning of ith frame, $\mathcal{F}_i$, based on the past frame's thermal profile estimations, *Aquilo* predicts the temperature of each of the antennas as if they are scheduled for data transfer in $\mathcal{F}_i$. Based on the predictions, *Aquilo* sorts a list of antenna schedules and selects the one with the lowest peak temperature. Each antenna transfers data for a limited time only, which we call a *slot* s; and, at the beginning of every slot s_k, *Aquilo* probes and switches to the new antenna as per the selected schedule. In parallel, *Aquilo* constantly monitors the temperature of both the active and idle antennas, and update a running estimation of their thermal profiles. At the end of $\mathcal{F}_i$, *Aquilo* uses the newly estimated thermal profiles for scheduling in the next frame, $\mathcal{F}_{i+1}$; and the above process repeats. Fig. 7(c) shows an illustrative run-time example of *Aquilo*.

A key challenge is to identify the suitable lengths for the slot and frame. *Aquilo* will incur high probing and frequent switching overheads if the slot length is too small. But the scheduled antenna will likely change performance within one slot if the length is too large. Besides, slot length and number of slots determine the frame length (slot length $\times$ number of slots). A larger frame length means better thermal profiles estimation; but that profile may not be usable for the next frame since the estimation may become stale. This staleness is due to the temporal variations of different factors, such as the surrounding temperature and the location of the user's hand or body. Also, if the frame length is too large with many slots, *Aquilo* will incur a high computational overhead for thermal profile estimation. For example, if there are 4 mmWave antennas and 5 slots per frame, then the number of possible schedules is 1,024; with only 8 slots per frame, it increases to 65,536.

Non-Adjacency Criteria One way *Aquilo* reduces this huge list of schedules is by leveraging a simple heuristic: Avoid scheduling the same antenna in back-to-back time slots. We call it a *Non-Adjacency Criteria* (NAC). The peak temperature attained by a schedule where the same antenna is never scheduled back-to-back is always lower than any other schedule; this is because an antenna starts to cool down immediately after it stops transferring data (Section 3). This heuristic can reduce the number of schedules exponentially. In the previous example, the number of schedules with NAC is 8748, an almost 8$\times$ reduction from 65,536. In Section 6, we have also verified the performance of this heuristic empirically in reducing the peak temperature.

Removing only the last scheduled antenna from the candidate set is reasonable in a system with few antennas, such as two to four. However, in a system with more antennas, it might be more efficient to remove up to two, three, *etc.*, antennas from the candidate set. To achieve better thermal mitigation, *Aquilo* would like to increase the idle time for each antenna in between its transmission phases. Thus, the antennas scheduled in the past few slots have a low probability of being scheduled in the next slot in a thermally optimal scheme and could be eliminated from the candidate set. This can significantly reduce the search space of candidate schedules even further.

For instance, in a system with six antennas, if the schedule for the first four slots is *{2, 3, 6, 4}*, and *Aquilo* stipulates that the last three scheduled antennas cannot be scheduled in the slot, then the set of possible antennas which can be scheduled in slot 5 is reduced from *{1, 2, 3, 4, 5, 6}* to *{1, 2, 5}* (note that antennas *3, 4, 6* are eliminated). So, if there are η slots in one frame, the number of possible schedules is reduced from $6\times5^{\eta-1}$ to $6\times5\times4\times3^{\eta-3}$. For example, for ten slots, the absolute reduction is from 11,718,750 candidate schedules to 262,440. In general, if the system has N antennas to be scheduled in η slots, and in each slot, a maximum κ of the previously scheduled antennas are to be removed, then

the possible number of schedules is: $\prod_{d=0}^{d=\kappa-1}(N-d)\times(N-\kappa)^{\eta-\kappa}$. Fig. 8 shows the effect of reduction in search space for a system with 8 antennas and 7 slots.

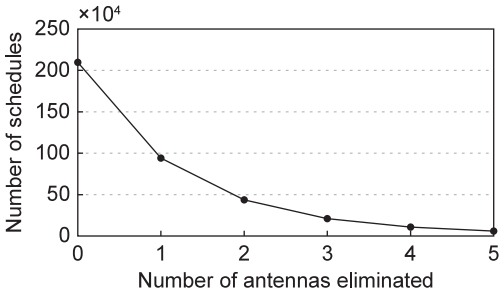

Fig. 8. Effect of increasing the number of antennas eliminated from the candidate sets in a system with 8 antennas and 7 slots.

Example Schedule and Performance Figs. 9(a–d) show example of antenna schedule, system's throughput, and peak temperature under *Aquilo* and a throughput optimization scheme.

We set up our smartphone at 1 m away from the AP and transfer data at a peak rate continuously for 90 s; the channel remains in a near-static condition. We simulated a 4 antenna mmWave system with slot length 10 s. and frame length 90 s. For visual clarity, we averaged the system's throughput and temperature every 10 s window. Figs. 9(a–b) show the scheduled antenna numbers and the corresponding throughput. Clearly, throughput optimization achieves better throughput. However, it suffers from a very high system's temperature since it only selects two out of four antennas with the best throughput; its peak temperature reaches 72°C. But *Aquilo* selects the antenna to minimize the system's temperature. The peak temperature never crosses beyond 55°C — *i.e.*, 17°C temperature reduction sacrificing only 6% average throughput.

Algorithm for Multi-Antenna Scheduling Algorithm 1 formally describes the antenna scheduling process. At the beginning of every frame $\mathcal{F}_i$, *Aquilo* gathers the thermal profile for all antennas scheduled in frame $\mathcal{F}_{i-1}$; for each schedule s_j, it can pre-compute the peak temperature and create a list of sorted schedules, S. *Aquilo* then probes the first antenna from the first schedule in the list. Due to variable connectivity, this antenna may fail to establish a link, *Aquilo* continuously probes the other antennas in the first schedule, removes the unusable antennas, and updates the list of schedules. Once *Aquilo* is able to establish a link, it starts the data transfer, adapting its beam direction and bit-rate, and updating its thermal profiles. Finally, after *Aquilo* completes every slot scheduled in the current frame, it uses the new thermal profile estimation in the next frame, $\mathcal{F}_{i+1}$.

4.3. Latency for antenna probing and switching

Since at the beginning of every slot *Aquilo* probes the selected antenna, it may incur additional latency; this is because, during the antenna probe, no useful data transfer occurs. Using state-of-the-art fast beam alignment protocols, we can reduce the latency below sub-ms [48,53], but the number of switches and total latency may be higher than other "temperature-unaware" schemes. However, the slot length in *Aquilo* is much longer than beam alignment latency; therefore, the relative overhead is very low. Next-generation mmWave standards also use multiple RF chains so that many antennas can be active simultaneously and operate independently [34–36,78]. (An RF chain consists of amplifiers, modulators, filters, PLLs, ADC/DACs, etc., and processes the wireless signal.) So, *Aquilo* can continue data transfer with the active antenna, while simultaneously probing and preparing to switch to another antenna, incurring no additional latency. We have also evaluated the effect of antenna switching on *Aquilo*'s network delay performance in Section 6.

4.4. Integrating Aquilo with IEEE 802.11ad and 5G NR

Aquilo can be integrated with IEEE 802.11ad/ay or 5G NR COTS devices seamlessly. *Aquilo*'s slots and frames span multiple beacon intervals in IEEE 802.11ad (100 ms.) or multiple radio frames in 5G NR (10 ms.). Throughout a beacon interval or radio frame duration, *Aquilo* can follow the standard protocols for aligning the beam directions and transferring data with the active antenna. At the beginning of each slot, instead of initiating a beam alignment from the current antenna per the standard, *Aquilo* initiates the probing and switches to the appropriate antenna as per its schedule. Since this probe and switch happens after every slot, that spans several beacon intervals, the relative cost is very low. Besides, in the future, *Aquilo* can be integrated into the standard devices to provide *device temperature as a Quality of Service*, by enabling flexible guarantees on the peak temperature of the mmWave devices.

Algorithm 1 *Temperature-aware antenna scheduling*

Frame, $\mathcal{F}_i$; In: Thermal profiles $Th_k^{(i-1)}$ for all k antennas in $\mathcal{F}_{i-1}$; Out: New thermal profile Th_k^i;
For every schedules, $s_j = \{a_1, a_2, a_3, \ldots, a_\eta\}$ $\forall j$
  $PT_{s_j} \leftarrow max\{PT(a_1), \ldots, PT(a_\eta)\}$; PT: Peak Temp.;
endFor
Sorted list of schedules: $S = \{s_1, s_2, s_3, \ldots, s_N\}$, *s.t.*, $PT_{s_1} \leq PT_{s_2} \leq \ldots \leq PT_{s_N}$;
Initialize the current schedule: $\alpha_i = \{\}$;
For each slot, $k \in \{1, \ldots, \eta\}$
  Scheduled antenna, $SA \leftarrow s_1(a_k)$; $link \leftarrow probe(SA)$;
  while *link* is *false*
    Remove all s_j which starts with antenna SA; $S \leftarrow S \setminus s_j$, $\forall s_j$, *s.t.*, $SA = s_j(a_k)$;
    $SA \leftarrow s_1(a_k)$; $link \leftarrow probe(SA)$;
  endwhile
  $\alpha_i \leftarrow \{\alpha_i | SA\}$; $switch(SA)$; Update thermal profile for SA: Th_{SA}^i;
endFor

5. Implementation

We implement and evaluate *Aquilo* by collecting throughput and temperature measurements from our COTS testbed consisting of a NETGEAR X10 AP and ASUS ROG smartphone (Fig. 2[a]). While there are a few other 60 GHz smartphones in the market currently, ASUS ROG has a similar specification and capability as the others. The mmWave antenna on both the AP and smartphone can generate up to 64 transmit and receive beams in 3D. Under a multi-antenna device, each antenna can generate its own set of 64 beams; so when *Aquilo* switches to a new antenna, it selects the antenna's beam with the strongest link to the AP. Under mobility and blockage, *Aquilo* steers the selected antenna's beam to support reliable connectivity. Both the AP and smartphones can measure the wireless bit-rate, link throughput, and temperature of its mmWave antenna every 1s. Throughout our evaluation, we consider IEEE 802.11ad [31] as the underlying standard, and follow its beacon structures, beam alignment process, and antenna switching overheads. We collect measurements for various data transfer periods ranging from 5 to 100 s. Between two measurements, the set up was allowed to cool down to its idle temperature so that all measurements have a common baseline. To expedite the cooling process, we also used an external USB fan.

Since the COTS devices currently do not have multiple mmWave antennas, we run trace-driven simulations. *First*, for each experiment, we collect measurements by fixing the smartphone's orientation. *Then*, we rotate it to 4 different orientations *w.r.t.* the AP to create antenna's different positions. *Finally*, we combine the measurements to emulate a setup with 4 mmWave

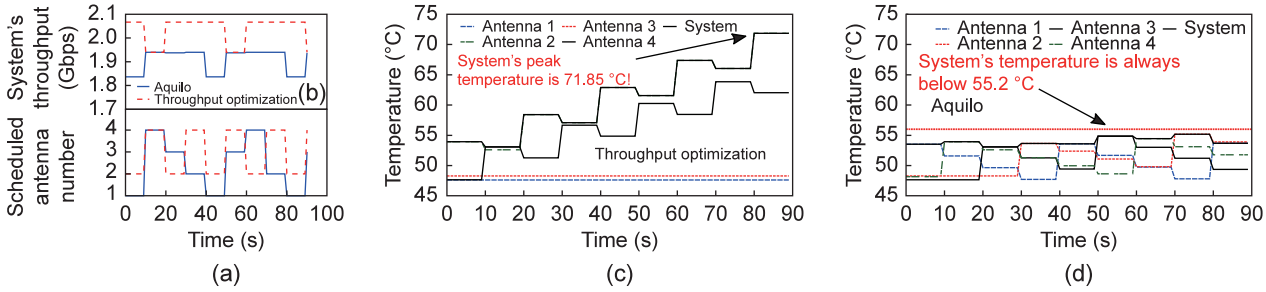

Fig. 9. Example results for a near-static 90 s. of data transfer under *Aquilo* and throughput optimization. (a) Scheduled antenna numbers; (b) System's throughput; (c–d) Temperature changes over time. Peak temperatures under throughput optimization and *Aquilo* are 71.85°C and 55.20°C, respectively.

antennas. However, a challenge with this emulation is that it does not capture the effects of the user's device handling, where she may obstruct one or many antennas occasionally. To simulate these effects, we introduce random blockage as a probability of antenna producing zero throughputs; we also vary this probability to simulate the different intensity of obstructions.

6. Evaluation

We evaluate *Aquilo* by considering performance along two dimensions: system throughput and peak temperature. We will show the following in our evaluation: (1) *Aquilo* reaches a median peak temperature 1°C above the best case temperature while sacrificing 9.8% of throughput under various static conditions. (2) Under various degrees of obstructions, *Aquilo* can approximate the temperature-optimal scheme, with 1.1°C to 5.4°C differences, while maintaining above 1.77 Gbps in more than half of the cases. (3) While *Aquilo* needs to trade-off temperature-optimal antennas under higher throughput requirements, it still outperforms a heuristic-based scheme by 3.9°C to 7.4°C. (4) A larger frame length has poorer performance and higher computational burden, and a smaller slot length has higher probing and switching overheads; in practice, 1 s. slot and 10 s. frames perform well. (5) More antenna choices in a device allow *Aquilo* for better thermal mitigation without affecting the throughput significantly; the increase in search space can be efficiently handled by extending the *Non-Adjacency Criteria* to remove more than one antenna from candidate schedules. (6) Under a *trigger-temperature* mode, *Aquilo* achieves fairly well temperature minimization and strikes the right balance between the temperature and throughput. (7) Under mobility, *Aquilo* shows near-optimal performance, with 0.5°C median difference from the optimal, even when the link demands 1 Gbps throughput. (8) Finally, for field trials with FTP and VR, *Aquilo* provides a similar quality of experience as the best throughput scheme while keeping the temperature at near-optimal level.

6.1. Compared schemes

We compare *Aquilo* with the following four schemes:

Best Case Temperature An "oracle" scheme that finds the minimum system temperature under an experimental condition, considering all possible blockage conditions, temperature changes, and antenna performance ahead of time but after the fact. Although an impractical scheme, this provides the lower bound on the peak temperature.

Random Scheduling with Non-Adjacency Criteria The simplest way to schedule is to *randomly* select an antenna, ensure that it meets the performance requirement, and start data transfer. Since switching an antenna from active to idle state reduces its temperature immediately, we consider a random selection, but ensure that the same antenna is never assigned in adjacent time slots. It allows shuffling so that the peak temperature does not grow steadily.

Random Scheduling This scheme selects antennas randomly, but without the non-adjacency criteria, and ensure they meet the application's performance requirement.

Throughput Optimization It tries to maximize only the throughput, without regard to any knowledge of temperature. It selects the schedule with the highest throughput and determines the upper bound of a throughput maximization scheme.

6.2. Microbenchmark results

Performance under Static Conditions We first measure *Aquilo*'s effectiveness under static conditions. We choose 200 static links, each with 100 s. duration, and estimate the best case peak temperature. To estimate the ground truth of maximum throughput, we also run the throughput optimization scheme. In addition, we run the random scheduling with and without the *Non-Adjacency Criteria* (NAC). While pure random scheduling captures the average case temperature behavior, random scheduling with NAC evaluates a simple thermal mitigation technique: Allow periodic idleness by avoiding adjacent time slots. Finally, we compare the throughput and peak temperature results from *Aquilo*.

Figs. 10(a–b) show the empirical Cumulative Distribution Function (CDF) of the system's peak temperature and throughput. The median of best case peak temperatures is 53.64°C. The throughput optimization shows the worst case temperature performance; its median peak temperature is beyond 67°C, even worse than random scheduling. This is because it only maximizes the throughput and is "temperature-unaware". Furthermore, a simple heuristic of NAC effectively improves the median peak temperature by more than 3.7°C from the random selection. *Aquilo* outperforms both random scheduling and throughput optimization in terms of temperature; its median peak temperature is just about 1°C above the best case. Compared to the throughput optimization, *Aquilo* sacrifices only $\sim$ 9.8% throughput, but in more than half of the cases, *Aquilo* reduces the peak temperature by approximately 12°C. This is an appreciably large reduction, and the temperature is contained to within 4°C of the idle state.

Effect of Environmental Dynamism We evaluate *Aquilo*'s performance under environmental dynamism. A dynamic environment is the one with moving people, objects around mmWave link, and user's handling of the device, like orientation changes, partial or complete obstructions of one or more antennas, *etc.* We have modeled these events as the varying probability of blockage: For example, if the probability is 0.2, then an antenna can successfully establish a link towards the AP only 80% of the time. Figs. 10(c–d) show the peak temperature and throughput results. The temperature performance of *Aquilo* degrades with increasing blockage probability; under 0.5 probability, the median

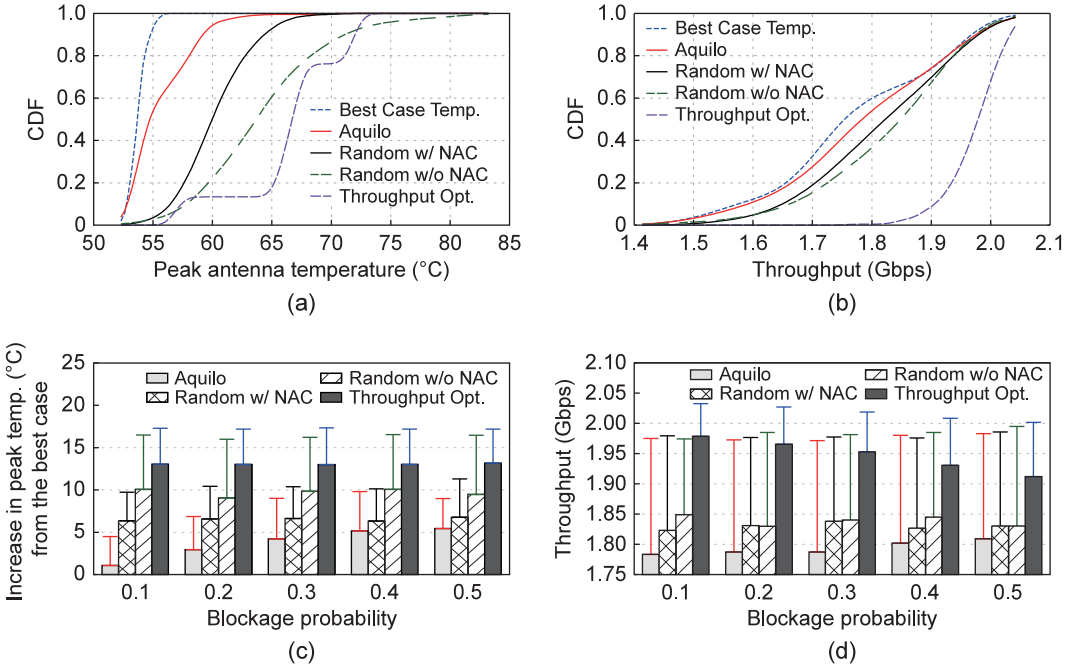

Fig. 10. Empirical CDFs from different schemes across various static conditions: (a) Peak antenna temperature; and (b) System's throughput. Effect of increasing environmental blockage on the performance: (c) Increase in peak antenna temperatures from the best case; and (d) System's throughput. The error bar shows 90th percentile value.

peak temperature is more than 5°C above the optimal. This is expected: As the blockage probability increases, *Aquilo* has limited choices and flexibility in scheduling antennas with better thermal mitigation, and thus, resort to whatever is available for continued operation. A blockage probability of 0.5 is an extreme case and hence will generate a much worse network performance. The result shows that even under such an extreme case, the antenna temperature is just 5°C above the best case.

Delay Performance *Aquilo* may affect the network delay performance, since it achieves thermal mitigation with antenna probing. To find out the additional latency under *Aquilo*, we simulate it in a device with 4 antennas, each with 64 beams, in IEEE 802.11ad standard network under various environmental dynamics. We use the standard beam alignment latency [46,48], and simulated each blockage configurations for 1000 instances. Fig. 11 shows the additional latency from *Aquilo* under a 100 s time window. As expected, the latency increases with higher blockage probability; this is because, under higher blockage probability, *Aquilo* may need to switch antennas more often, not only for reliable connectivity, but also for better thermal performance. However, even under an intense blockage occurrence with 0.5 probability, the total additional latency, averaged over 1000 instances, is 36.36 ms — an overhead of less than 0.04%, which is tolerable in practice.

Performance under Different Throughput Requirement To support the stringent throughput requirements of applications, like wireless AR/VR, *Aquilo* may need to trade-off temperature-optimal antennas for throughput performance. Algorithm 1 will be able to schedule only those antennas that guarantee performance at or above the requirements. Thus, a higher minimum throughput requirement will degrade the thermal mitigation effectiveness. Fig. 12(a) illustrates this point; at higher throughput requirement, *Aquilo*'s performance deviates significantly from the best case temperature and moves closer to random scheduling with NAC. Still, *Aquilo*'s effectiveness is well maintained when the throughput requirement is 1.4 Gbps or below.

Effect of Different Frame Lengths We evaluate *Aquilo*'s performance in selecting optimal schedules under different frame lengths, but fixed slot lengths. Fig. 12(b) shows the difference in median peak temperatures between the best case and *Aquilo*. We have three observations. *First*, the effectiveness of *Aquilo* drops with higher frame length; this is because *Aquilo* uses past frame's thermal profile estimation to schedule antennas in the next frame, which is more likely to be ineffective if the frame lengths are too large. *Second*, the number of candidate schedules increases with longer frame duration (shown with boxed numbers in Fig. 12[b]), and thus, in practice, *Aquilo* will have higher computational overhead. *Finally*, irrespective of the frame length, *Aquilo*'s median peak temperature still stays within 2°C above the optimal.

Performance under Mobility We now evaluate *Aquilo*'s performance under mobility. We collect the trace data for various mobility durations between 10 to 50 s., and measure the difference in peak temperatures between the best case and *Aquilo*. We consider two factors in our mobility evaluation: variable slot length; and variable minimum throughput requirement. Fig. 12(c)

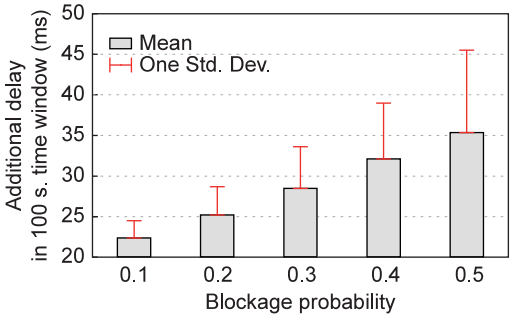

Fig. 11. *Aquilo* delay within a window of 100 s. for different environmental conditions; Error bar is one standard deviation. The average overhead is less than 0.04%.

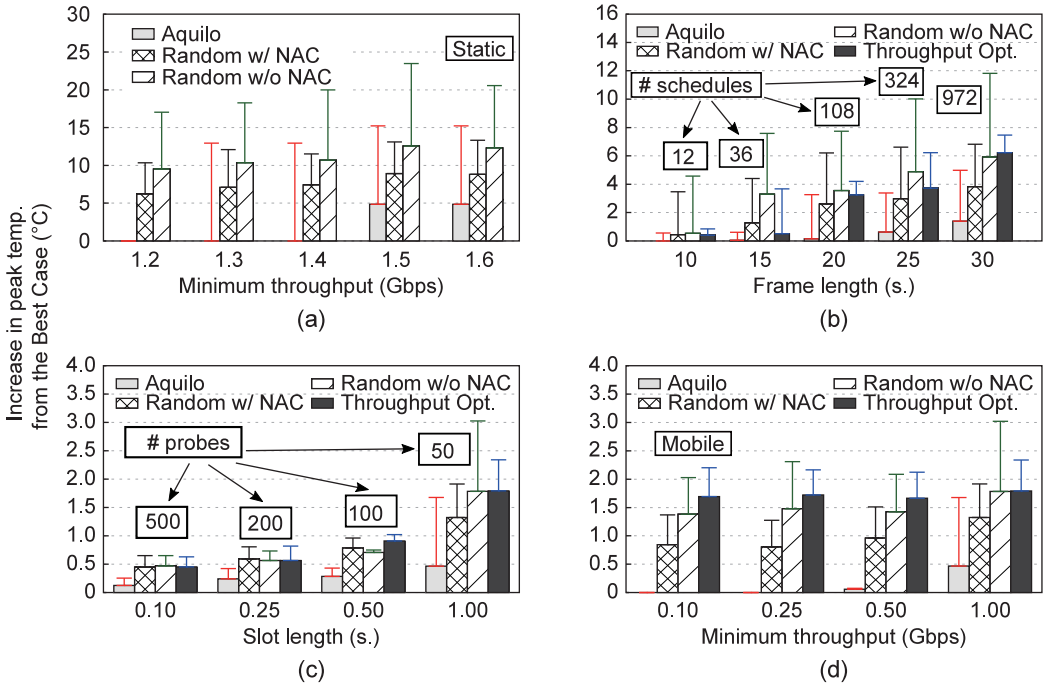

Fig. 12. Increase in peak temperature from the best case. Under static cases, effects of: (a) Minimum throughput requirement; and (b) Frame lengths on peak temperature and # of schedules. Under mobile cases: (c) Effect of slot lengths, on peak temperature and # of probes, when min. throughput is 1 Gbps; and (d) Effect of minimum throughput, when slot length is 1 s.

shows that the effectiveness of *Aquilo* increases with smaller slot lengths. This is intuitive: A smaller slot length, such as 0.1 s., allows fast switching between antennas and more frequent schedule updates. However, the total number of probings will be very high (shown with boxed numbers in Fig. 12[c]). For example, over 50 s, with a slot length of 0.1 s., the number of probings will be 500. Such a high overhead may not be worth the extra gain in temperature performance. Nonetheless, a slot length of 1 s. enables reasonable thermal mitigation; and, the median peak temperature is only $\sim 0.5°C$ above the optimal.

Fig. 12(d) shows the effect of the minimum throughput requirement of mobile links. *Aquilo* achieves near-optimal performance when the requirement is at or below 0.5 Gbps; but the performance deviates from the optimal, under stringent 1 Gbps requirement. The performance is also relatively poor compared to the static cases (Fig. 12[a]), since under mobility, *Aquilo* needs to trade-off temperature-optimal antennas for throughput more often than static cases. Still, the median peak temperature is only 0.5°C above the best case, even with 1 Gbps throughput requirement.

6.3. Scalability to varying number of antennas

A four-antenna architecture is a reasonable design choice for a smartphone, but depending on the form factor, a device can accommodate a variable number of antennas. For example, a smartwatch may have only two antennas, but a VR/AR headset may have as many as eight antennas. We now evaluate *Aquilo*'s scalability by running experiments with varying number of antennas in a system: 2, 4, 6, and 8. We compare *Aquilo* with the best case temperature and throughput optimization to show that it can achieve the right balance between the throughput and the temperature, irrespective of the number of antennas.

Figs. 13(a–b) show the throughput and temperature results under different numbers of antennas. For each case, we evaluate the schemes with 7 scheduling slots, a mean slot length of 10 s and a blockage probability of 0.2. A two antenna architecture offers a very narrow space of choices: including the *Non-Adjacency Criteria* restricts the possible number of antenna schedules to only two, *i.e.*, either transmission begins with the first antenna and then alternates between the first and second, or transmission begins with the second antenna and then alternates between the second and first. The number of possible schedules with 8 antenna, on the other hand, is manifold higher due to massively large number of possible combinations. It is expected that temperature minimization will improve with an increasing number of antennas due to the expansion of choice space. This is illustrated in Fig. 13(a), which shows that the peak temperature under the best case scheme decreases from around 62°C to around 51°C in the median case from 2 antennas to 8 antennas. The temperature under *Aquilo* also decreases almost in tandem, keeping a difference of around 1°C than the best case. The throughput optimization scheme shows no particular trend in peak temperature with an increasing number of antennas, which is expected since this is not a temperature-aware scheme.

The throughput optimization (Fig. 13[b]) shows an increasing trend in throughput, while both the best case temperature and *Aquilo* show a decreasing trend. For the throughput optimization, the search space increases with more antennas, and hence, throughput can be optimized further. The median throughput increases from 1.93 Gbps to 1.99 Gbps. For the best case temperature and *Aquilo*, the antennas which would have higher throughput will be sacrificed more due to high temperature and scheduled much later than in a system with fewer antennas. Even so, the median throughput remains above 1.65 Gbps, which is greater than the requirement in even the most data-hungry applications [79].

There is, however, a related challenge due to increasingly more antennas in the system. Due to the massively large number of possible schedules, the search space is extremely large. We can reduce this search space by extending the *Non-Adjacency Criteria*, as discussed in Section 4.2. Fig. 14(a) shows that removing more antennas from the choice space does not increase the median

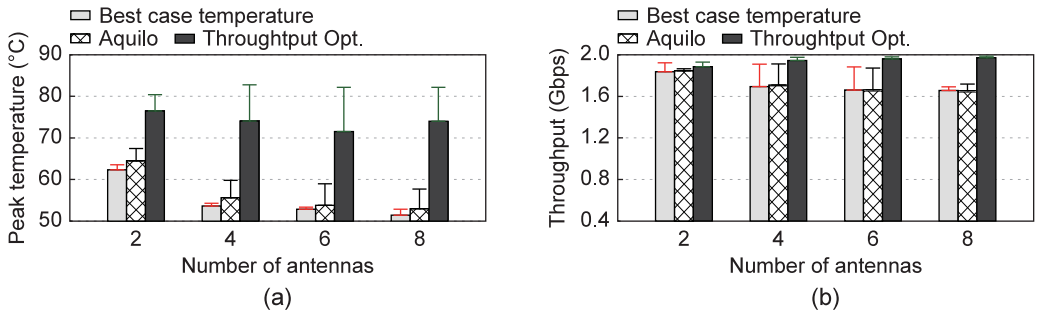

Fig. 13. Performance change with an increasing number of antenna: (a) Peak temperature; (b) Throughput.

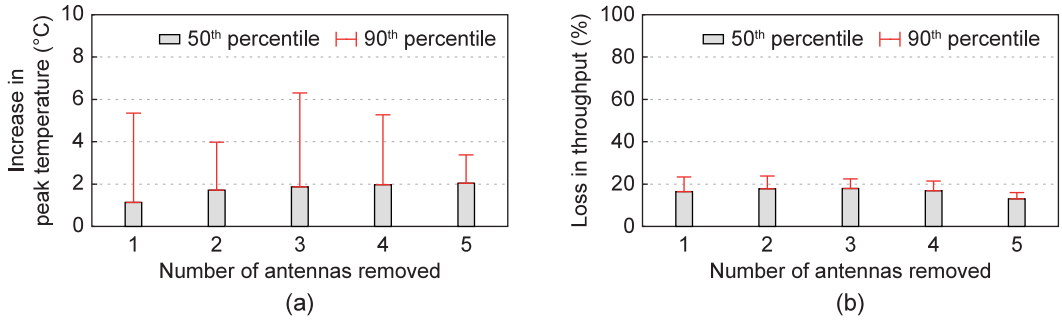

Fig. 14. Adverse effects, *i.e.*, (a) Increase in peak temperature; and (b) Loss in throughput of increasingly more antennas removed.

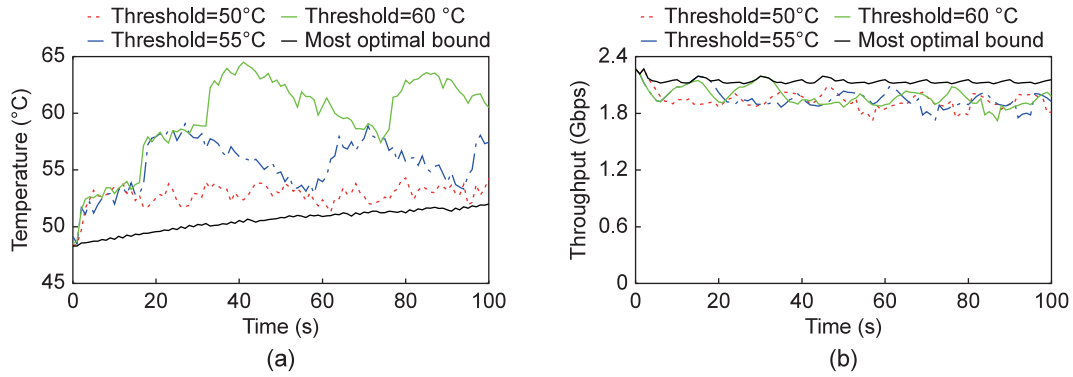

Fig. 15. Peak temperature and system throughput under *Aquilo* with different threshold temperatures compared with oracle schemes.

peak temperature above the best case temperature by more than 2°C in a system with 8 antennas and 7 slots. The loss in median (Fig. 14[b]) throughput is consistently below 20% when the last five scheduled antennas are removed from the choice space and do not affect the data-hungry applications [79].

6.4. Trigger Temperature for Aquilo

So far, we have assumed that *Aquilo* runs continuously from the time of device boot-up. We would also like to investigate the scenario when *Aquilo* allows the best throughput antenna to transmit from the beginning until the system temperature crosses a pre-set threshold, so *Aquilo* is triggered only when the temperature threshold is crossed. This flexibility further allows the system designer to prioritize throughput over temperature under certain application contexts.

We carry out this investigation on a system with 4 antennas, 10 slots, and blockage probability of 0.2. We use three temperature thresholds: 50°C, 55°C, and 60°C. The transmission begins with the antenna with the best instantaneous link at the time of boot-up and continues with this antenna until the temperature crosses the threshold. *Aquilo* is then triggered to optimize the temperature performance. Fig. 15(a) shows that the temperature minimization is achieved fairly well for the threshold 50°C, and system temperature under *Aquilo* is always within 5°C of the most optimal bound. For 55°C threshold, the temperature can rise up to 58°C, but it is not continuously so warm, since *Aquilo* kicks in to reduce the temperature. For 60°C threshold, however, the temperature can rise further and can reach up to 65°C — almost 15°C increase from the most optimal bound. Fig. 15(b) further shows that the system throughput remains stable at around 2.1 Gbps (±0.1 Gbps), with the 60°C temperature threshold, which is comparable to the most optimal bound (black line). Even under different temperature thresholds, the throughput consistently remains above the 1.8 Gbps mark, which is more than the minimum throughput requirement of many data-hungry applications, such as AR, VR, or tactile internet [79].

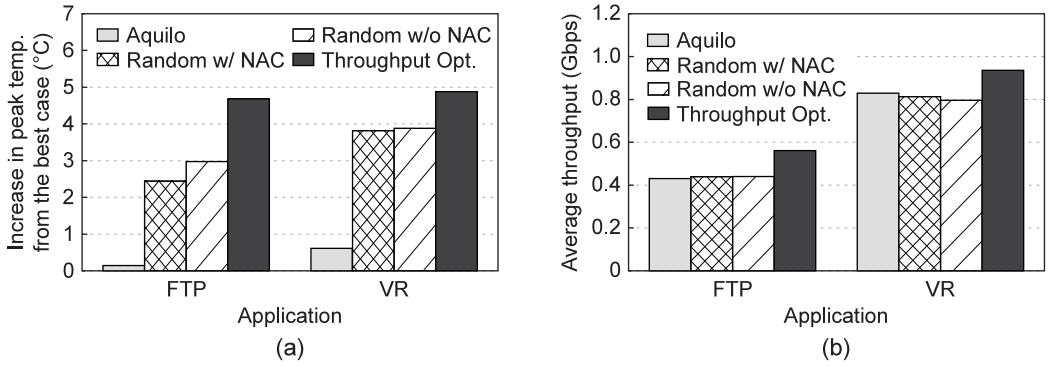

Fig. 16. Performance in field trial applications: (a) System's peak temperature; (b) Application's throughput.

6.5. Field trials

Finally, we evaluate *Aquilo* on real applications; we use previous mobility set up and collect temperature and throughput trace data while running two applications: FTP; and Virtual Reality (VR) gaming. *First*, we set up an FTP server on the AP, and have the smartphone download a 2 GB file. *Then*, we set up a gaming server on a PC [80] that streams real-time data (video, voice, control) over the smartphone via the 60 GHz mmWave link; we also used a Google cardboard [81] to set up a smartphone VR. However, all our field trials are limited to 1 Gbps maximum throughput, since this is highest possible transfer rate with the Ethernet port on the PC. Figs. 16(a–b) show the average throughput and peak temperature performance. For both the applications, *Aquilo* maintains the required throughput performance while reducing median peak temperature within 0.75°C of the best case temperature scheme. In summary, *Aquilo provides a similar quality of experience as the best throughput scheme, while simultaneously reaching near-optimal device temperature.*

7. Conclusion

In this paper, we present the first-of-a-kind study on mmWave thermal characterization; the study reveals new challenges and opportunities to keep IEEE 802.11ad and 5G NR devices cool. Based on the measurement insights, we propose *Aquilo*, the first temperature-aware, scalable multi-antenna scheduler for mmWave devices and networks. We use testbed experiments from COTS mmWave devices to demonstrate *Aquilo*'s effectiveness in maintaining link performance while reducing temperature substantially. In the future, we plan to continue investigating mmWave performance fluctuations under high device temperature with more fine-grained measurements, and characterizing thermal performance under different applications and use cases, both indoor and outdoor; besides, we plan to design, implement, and evaluate a real-time *Aquilo* on commercial multi-antenna mmWave devices. Overall, we believe, our research on mmWave device cooling helps allay concerns in some quarters about health effects of 5G and accelerates its deployment broadly.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors were partially supported by the NSF (CNS-1910853, MRI-2018966 and CAREER-2144505)

References

[1] IEEE P802.11 - Task Group ay, Status of project IEEE 802.11ay, 2020, http://www.ieee802.org/11/Reports/tgay_update.htm.
[2] 3GPP: A. Global Initiative, The mobile broadband standard, 2021, http://www.3gpp.org/release-17.
[3] E. Dahlman, S. Parkvall, J. Skold, 5G NR: The Next Generation Wireless Access Technology, Elsevier, 2018.
[4] Qualcomm Incorporated, Mobilizing mmWave with 5G NR, 2020, https://www.qualcomm.com/invention/5g/5g-nr/mmwave.
[5] Google LLC, Google soli, 2019, https://atap.google.com/soli/.
[6] Qualcomm Incorporated, Qualcomm announces 5G NR mmWave prototype to accelerate mobile deployments for smartphones, 2017, https://bit.ly/3Ciu1lA.
[7] AsusTek Computer Inc., ASUS ROG phone, 2018, https://www.asus.com/us/Phone/ROG-Phone/.
[8] Apple Inc., US patent app: Electronic devices having millimeter wave ranging capabilities, 2019, https://bit.ly/2k0rsSr.
[9] Patently Apple, Apple files patents for future iPhones with gas sensors & 5G millimeter wave antennas for apple watch, 2019, https://bit.ly/3megLJb.
[10] TPCAST U.S., Inc., TPCAST: Unleash the VR world, 2019, https://www.tpcastvr.com/.
[11] HTC, HTC VIVE PRO, 2019, https://www.vive.com/us/product/vive-pro/.
[12] VentureBeat, SONY VR patent points to a pricier but wireless PSVR2, 2019, https://bit.ly/3ClEhd2.
[13] Intel Corporation, Intel wireless gigabit for VR, 2019, https://intel.ly/3B8eOCg.
[14] S.K. Saha, D.G. Malleshappa, A. Palamanda, V.V. Vira, A. Garg, D. Koutsonikolas, 60 GHz indoor WLANs: Insights into performance and power consumption, J. Wirel. Netw. 24 (7) (2018).
[15] S.K. Saha, T. Siddiqui, D. Koutsonikolas, A. Loch, J. Widmer, R. Sridhar, A detailed look into power consumption of commodity 60 GHz devices, in: IEEE WoWMoM, 2017.
[16] ExtremeTech, 5G modems and phones literally can't handle the heat of summer weather, 2020, https://bit.ly/3Gj4tqV.
[17] HackADay, 5G power usage is making phones overheat ..., 2019, https://bit.ly/30UqIDf.
[18] M.S. Saadat, S. Sur, S. Nelakuditi, A case for temperature-aware scheduler for millimeter-wave devices and networks, in: IEEE 28th International Conference on Network Protocols, ICNP, 2020.
[19] M.S. Saadat, S. Sur, S. Nelakuditi, Bringing temperature-awareness to millimeter-wave networks, in: ACM MobiCom, 2020.
[20] A. Keykhosravi, M. Neamatshahi, R. Mahmoodi, E. Navipour, Radiation effects of mobile phones and tablets on the skin: A systematic review, Adv. Med. (2018).
[21] O.P. Gandhi, A. Riazi, Absorption of millimeter waves by human beings and its biological implications, IEEE Trans. Microw. Theory Tech. 34 (2) (1986).
[22] Qualcomm Incorporated, Breaking the wireless barriers to mobilize 5G NR mmWave, 2019, https://bit.ly/3pCWb7w.
[23] Qualcomm Incorporated, 5G mmwave radio design for mobile, 2017, shorturl.at/bfiFG.
[24] D. Brooks, M. Martonosi, Dynamic thermal management for high-performance microprocessors, in: International Symposium of High-Performance Computer Architecture, HPCA, 2001.
[25] K. Skadron, Hybrid architectural dynamic thermal management, in: IEEE Proceedings Design, Automation and Test in Europe Conference and Exhibition, 2004.
[26] A. Kumar, L. Shang, L.-S. Peh, N.K. Jha, System-level dynamic thermal management for high-performance microprocessors, IEEE Trans. Comput.-Aid. Des. Integr. Circuits Syst. 27 (2008).

[27] A. Merkel, F. Bellosa, A. Weissel, Event-driven thermal management in SMP systems, in: Workshop on Temperature-Aware Computer Systems, 2005.
[28] H. Jung, M. Pedram, Stochastic dynamic thermal management: A Markovian decision-based approach, in: International Conference of Computer Design, 2007.
[29] NETGEAR, Inc., Nighthawk X10 smart WiFi Router, 2017, https://www.netgear.com/landings/ad7200/.
[30] S. Sur, Thermal characterization data and multi-antenna simulator code for 60 GHz millimeter-wave devices, 2021, https://syrex.cse.sc.edu/awards/nsf1910853.
[31] IEEE Standards Association, IEEE standards 802.11ad-2012, Amendment 3: Enhancements for very high throughput in the 60 GHz band, 2012, goo.gl/s7HzC2.
[32] R. Zhao, T. Woodford, T. Wei, Q. Kun, X. Zhang, M-Cube: A millimeter-wave massive MIMO software radio, in: Proc. of ACM MobiCom, 2020.
[33] S. Wang, J. Huang, X. Zhang, H. Kim, S. Dey, X-Array: Approximating omnidirectional millimeter-wave coverage using an array of phased arrays, in: Proc. of ACM MobiCom, 2020.
[34] S. Sun, T.S. Rappaport, M. Shafi, Hybrid beamforming for 5G millimeter-wave multi-cell networks, in: IEEE Conference on ComputerCommunications Workshop, 2018.
[35] J. Zhang, X. Yu, K.B. Letaief, Hybrid beamforming for 5G and beyond millimeter-wave systems: A holistic view, IEEE Open J. Commun. Soc. 1 (2019).
[36] R.W. Heath Jr., N. Gonzalez-Prelcic, S. Rangan, W. Roh, A. Sayeed, An overview of signal processing techniques for millimeter wave MIMO systems, IEEE JSTSP 10 (3) (2016).
[37] M.K. Samimi, T.S. Rappaport, 3-D statistical channel model for millimeter-wave outdoor mobile broadband communications, in: IEEE International Conference on Communications, ICC, 2015.
[38] T.S. Rappaport, E. Ben-Dor, J.N. Murdock, Y. Qiao, 38 GHz and 60 GHz angle-dependent propagation for cellular and peer-to-peer wireless communications, in: IEEE ICC, 2012.
[39] T.S. Rappaport, F. Gutierrez, E. Ben-Dor, J.N. Murdock, Y. Qiao, J.I. Tamir, Broadband millimeter-wave propagation measurements and models using adaptive-beam antennas for outdoor urban cellular communications, IEEE Trans. Antennas and Propagation 61 (4) (2013).
[40] T. Zwick, T.J. Beukema, H. Nam, Wideband channel sounder with measurements and model for the 60 GHz indoor radio channel, IEEE Trans. Veh. Technol. 54 (4) (2005).
[41] H. Xu, V. Kukshya, T.S. Rappaport, Spatial and temporal characteristics of 60-GHz indoor channels, IEEE J. Sel. Areas Commun. 20 (3) (2002).
[42] Y. Azar, G.N. Wong, K. Wang, R. Mayzus, J.K. Schulz, H. Zhao, F. Gutierrez, D. Hwang, T.S. Rappaport, 28 GHz propagation measurements for outdoor cellular communications using steerable beam antennas in New York city, in: IEEE ICC, 2013.
[43] T. Rappaport, F. Gutierrez, E. Ben-Dor, J. Murdock, Y. Qiao, J. Tamir, Broadband millimeter-wave propagation measurements and models using adaptive-beam antennas for outdoor urban cellular communications, IEEE Trans. Antennas and Propagation 61 (4) (2013).
[44] P.F.M. Smulders, Statistical characterization of 60-GHz indoor radio channels, IEEE Trans. Antennas and Propagation 57 (10) (2009).
[45] Y. Zhu, Z. Zhang, Z. Marzi, C. Nelson, U. Madhow, B.Y. Zhao, H. Zheng, Demystifying 60 GHz outdoor picocells, in: Proc. of ACM MobiCom, 2014.
[46] S. Sur, V. Venkateswaran, X. Zhang, P. Ramanathan, 60 GHz indoor networking through flexible beams: A link-level profiling, in: Proc. of ACM SIGMETRICS, 2015.
[47] S. Sur, X. Zhang, P. Ramanathan, R. Chandra, BeamSpy: Enabling robust 60 GHz links under blockage, in: USENIX NSDI, 2016.
[48] H. Hassanieh, O. Abari, M. Rodriguez, M. Abdelghany, D. Katabi, P. Indyk, Fast millimeter wave beam alignment, in: Proc. of ACM SIGCOMM, 2018.
[49] B. Li, Z. Zhou, W. Zou, X. Sun, G. Du, On the efficient beam-forming training for 60 GHz Wireless Personal Area networks, IEEE Trans. Wireless Commun. 12 (2) (2013).
[50] T. Nitsche, G. Bielsa, I. Tejado, A. Loch, J. Widmer, Boon and bane of 60 GHz networks: Practical insights into beamforming, interference, and frame level operation, in: ACM CoNEXT, 2015.
[51] S. Sur, I. Pefkianakis, X. Zhang, K.-H. Kim, WiFi-assisted 60 GHz wireless networks, in: ACM MobiCom, 2017.
[52] T. Wei, X. Zhang, Pose information assisted 60 GHz networks: Towards seamless coverage and mobility support, in: Proc. of ACM MobiCom, 2017.
[53] S. Sur, I. Pefkianakis, X. Zhang, K.-H. Kim, Towards scalable and ubiquitous millimeter-wave wireless networks, in: Proc. of ACM MobiCom, 2018.
[54] S. Jog, J. Wang, J. Guan, T.M.H. Hassanieh, R.R. Choudhury, Many-to-many beam alignment in millimeter wave networks, in: Proc. of USENIX NSDI, 2019.
[55] M. Park, P. Gopalakrishnan, R. Roberts, Interference mitigation techniques in 60 GHz wireless networks, IEEE Commun. Mag. 47 (12) (2009).
[56] C.-S. Sum, X. An, Z. Lan, T. Baykas, J. Wang, R. Funada, M.A. Rahman, H. Harada, S. Kato, A synchronization-frame-aided interference mitigation mechanism for millimeter-wave WPAN, in: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2009.
[57] S.K. Saha, S. Aggarwal, R. Pathak, D. Koutsonikolas, J. Widmer, MuSher: An agile multipath-TCP scheduler for dual-band 802.11ad/ac wireless LANs, in: Proc. of ACM MobiCom, 2019.
[58] Facebook Inc., Terragraph: Solving the urban bandwidth challenge, 2018, https://terragraph.com/.
[59] T.S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wanga, G.N. Wong, et al., Millimeter wave mobile communications for 5G cellular: It will work!, in: IEEE Access, Vol. 1, 2013.
[60] M. Elkashlan, T.Q. Duong, H.-H. Chen, Millimeter Wave Communications for 5G: fundmentals: Part 1, in: IEEE Communications Magazine, Vol. 52, 2014.
[61] M. Elkashlan, T.Q. Duong, H.-H. Chen, Millimeter wave communications for 5G: fundmentals: Part 2, in: IEEE Communications Magazine, Vol. 53, 2015.
[62] Y.L. Yong Niu, D. Jin, L. Su, A.V. Vasilakos, A survey of millimeter wave communications (mmWave) for 5G: Opportunities and challenges, Wirel. Netw. 21 (2015).
[63] X. Wang, L. Kong, F. Kong, F. Qiu, M. Xia, S. Arnon, G. Chen, Millimeter wave communication: A comprehensive survey, in: IEEE Communications Surveys & Tutorials, Vol. 20, 2018.
[64] T.S. Rappaport, Y. Xing, G.R. MacCartney, A.F. Molisch, E. Mellios, J. Zhang, Overview of millimeter wave communications for fifth-generation (5G) wireless networks - with a focus on propagation models, in: IEEE Transactions on Antennas and Propagation, Vol. 65, 2017.
[65] X. Zhou, J. Yang, M. Chrobak, Y. Zhang, Performance-aware thermal management via task scheduling, ACM Trans. Archit. Code Optim. 7 (1) (2010).
[66] M. Chrobak, C. Dürr, M. Hurand, J. Robert, Algorithms for Temperature-Aware Task Scheduling in Microprocessor Systems, Vol. 5034, Springer Publications, 2008.
[67] J. Liu, L. Zhong, Micro power management of active 802.11 interfaces, in: ACM MobiSys, 2008.
[68] G. Anastasi, M. Conti, E. Gregori, A. Passarella, 802.11 Power-saving mode for mobile computing in wi-fi hotspots: Limitations, enhancements and open issues, Springer Link Wirel. Netw. 14 (6) (2008).
[69] E. Rozner, V. Navda, R. Ramjee, S. Rayanchu, NAPman: Network-assisted power management for WiFi Devices, in: Proc. of ACM MobiSys, 2010.
[70] K. Flautner, S. Reinhardt, T. Mudge, Automatic performance setting for dynamic voltage scaling, in: Proc. of ACM MobiCom, 2001.
[71] X. Zhang, K.G. Shin, E-MiLi: Energy-minimizing idle listening in wireless networks, in: Proc. of ACM MobiCom, 2011.
[72] B. Gao, Z. Xiao, L. Su, D. Jin, L. Zeng, Energy-efficient idle listening scheme using 1 bit sampling in 60 GHz Wireless Local Area network, IET Commun. 9 (2015) 219–226.
[73] Qualcomm Incorporated, QCA9500, 2018, https://www.qualcomm.com/products/qca9500.
[74] FLIR Systems, Inc., FLIR ONE pro, 2021, https://www.flir.com/products/flir-one-pro/.
[75] S. Aggarwal, M. Ghoshal, P. Banerjee, D. Koutsonikolas, J. Widmer, 802.11Ad in smartphones: Energy efficiency, spatial reuse, and impact on applications, in: IEEE INFOCOM, 2021.
[76] S. Collonge, G. Zaharia, G.E. Zein, Influence of the human activity on wide-band characteristics of the 60 GHz indoor radio channel, IEEE Trans. Wireless Commun. 3 (6) (2004).
[77] D. Halperin, W. Hu, A. Sheth, D. Wetherall, Predictable 802.11 packet delivery fromwireless channel measurements, in: ACM SIGCOMM 2010.
[78] C. Kim, T. Kim, J.-Y. Seol, Multi-beam transmission diversity with hybrid beamforming for MIMO-OFDM systems, in: IEEE Globecom Workshops, Atlanta, GA, 2013.
[79] L. Xu, Context aware traffic identification kit (TriCK) for network selection in future HetNets/5G networks, in: 2017 International Symposium on Networks, Computers and Communications, ISNCC, 2017.
[80] ASUSTeK Computer Inc., ASUS ROG: Command and control: how to set up your own gaming server, 2016, https://bit.ly/3vJQoxU.
[81] Google, Google cardboard, 2020, https://arvr.google.com/cardboard/.