Unifying Address and Name Based Communication in Wireless Medium Access Control

Mohammed Elbadry, Fan Ye, Peter Milder
Electrical and Computer Engineering
Stony Brook University
New York, USA
{mohammed.elbadry, fan.ye, peter.milder}@stonybrook.edu

***Abstract*—Edge nodes deployed in edge environments (e.g., IoT) have two common use cases: *i)* access control and *ii)* disseminating data. An address based paradigm is suitable for sending access control commands to specific nodes, yet a name based one excels at filtering and disseminating data based on content regardless of who has the data. Current address based communication mandates grouping among nodes (e.g., using BSSID), incurring high overheads (e.g., periodic beacons); whereas name based discovery is much more efficient and flexible. However, the name based paradigm needs many assets possessed in the address based paradigm (e.g., frame rate adaptation). Such complementary strengths and needs call for unification of both paradigms to meet current edge environment needs. In this paper, we introduce a unified medium access control design that combines both paradigms. Our unified design eliminates the need for grouping in address based communication by a name based discovery protocol, enabling filtering based on address or data attributes. It leverages address based unicast rate adaptation to benefit name based communication through Transmission Configuration. Our experiments show the system's ability to discover neighbor addresses based on application attributes, and filter thousands of both name and address based entries efficiently. It reduces latency of name based communication by 30X, and loss rate from 10–20% to 0% through address based rate adaptation algorithm with reliability.**

***Index Terms*—Information-centric networking, NDN, Medium Access Control (MAC), rate adaptation**

I. Introduction

Edge computing is becoming more pervasive [1], [2]. Two categories of use cases are both common: data dissemination where a node requests data of certain content from non predetermined neighbors on the fly, and whichever have needed data reply; and access control, where a command is sent to a predetermined node for control actions (e.g., a specific light switch). In this paper we take the position that both name and address based communications have respective suitable use cases, and they must both be supported at the edge.

For example, consider an upcoming edge computing scenario where multiple distributed resource-constrained nodes perform identity validation based on visual facial recognition by a collection of cameras. Here, both data-centric and address based paradigms are needed: the control unit has to request video streams from nodes with cameras (data-centric), process the information, and (based on its internal database) signal a door unit to open or lock (address based).

Further, with newer standards coming and many medium access control (MAC) layer algorithms (e.g., MU-MIMO [3], frame rate adaptation [4], synchronized sleep [5]) developed for address-based communication, name-based communication needs to leverage these existing assets or adapt as much as possible to lower the development overhead and eliminate re-designing existing protocols for Information Centric Networking (e.g., NDN).

Thus, a Medium Access Control (MAC) layer that unifies both address and name based communication paradigms in wireless environments for the edge is needed. The unified MAC provides architectural abstraction so that underlying wireless algorithms can be leveraged for both communication paradigms (e.g., frame rate adaptation). Further, a unified MAC enables both paradigms to co-operate (e.g., use the name-based paradigm to discover nodes' addresses).

The most common NDN network layer is NFD [6] which leverages *dataname* to filter and route data and eliminate using addresses. With a data-centric MAC layer (e.g., V-MAC [7]), there is no need to discover networks or addresses to start communication as the network is flat. NDN is ideal for large data dissemination where it relies on a *consumer* sending an *interest* packet to request data, and then the *producer* proceeds to send the data. NDN is agnostic to the sender address: it does not care who sends the data and focuses solely on the content information. This enables a suite of capabilities (e..g, multicast, requesting data without knowing producer) that the address based stack is inherently unqualified to do elegantly. However, the current name based stack falls short in access control where commands must be sent to specific nodes with a given identity. Workarounds have been proposed by using persistent interest and others. However, because the nature of command is to control a specific, predetermined node, not to request some data that reside at non determined nodes to be discovered on the fly, such workarounds have multiple drawbacks (e.g., 2-way flow instead) and are fundamentally unfit for control.

Address based communication possesses many assets currently missing but needed in name based communication. For example, frame rate adaptation dynamically adjusts the transmission rate on a per frame basis to ensure the highest possible data rates under varying wireless reception conditions. The current address based communication paradigm uses the TCP/IP network layer which is ideal for reliable communication (i.e. unicast with 100% reliability) where a node needs to send a command to particular node. The most commonly used MAC mode for edge devices is ad hoc [8], [9]. Ad hoc must create a network (e.g., group address) before any communication can happen. Each node in the network sends a periodic announcement message (beacon) to share its address, announce supported rates, and synchronize an internal clock with every other node within the network. The use of periodic beacon causes multiple issues: *i)* high medium overhead as most of the nodes send them periodically; *ii)* group address is not necessary when discovery can be done based on data needs.

With both paradigms' advantages and limitations complementing each other, we build a unified MAC layer. Our MAC presents a flat network, supporting name and address based unicast communication with rate adaptation, and multicast with robust, low loss rate protocol. We design a discovery protocol that enables discovering neighbors' addresses based on attributes thus reusing the same discovery mechanism as name based communication and eliminating periodic announcement (i.e. beacons) and arbitrary grouping like ad hoc. Our unified MAC can leverage the frame rate adaption algorithm in address based communication to achieve adaptable, high data rates in name based communication when only one consumer exists. We build a *multicast initial rate selection* to leverage existing work to select the ideal rate for multiple consumers. Our experiments show decreasing latency of name based communication from 30s to 1s (30X) while retaining lower loss rate than using the name based stack (1–4% vs 0%).

In this paper we claim the following contributions:

- We design a unified name- and address-based MAC layer. We enable a *stable entry* structure to add addresses for address-based filtering within V-MAC's Lingering Encoding Table (LET) for an indefinite time. A *stable entry* allows for multiple address based communications to operate concurrently with name based communication. As an example, a node can send and receive address based packets (ad hoc), publish and subscribe name based frames (V-MAC), and operate in monitoring mode all concurrently without any switching, which is impossible using any existing address- or name-based radio.
- We design a neighbor address discovery protocol that can discover nodes with specific desired application level attributes (e.g., those having cameras with 1080p resolution).Thus we can eliminate the needs, thus overhead and complexities for forming, managing groups (e.g., beacons) yet still use addresses when predetermined destinations are known (e.g., access control)
- We design *transmission configuration* that maps data published by a producer to the consumer's address (obtained from interest received) to allow for unicast frame rate adaptation algorithm operation while doing name based transmission. We further develop a station aggregation method that allows for name based multicast communication to select the initial ideal rate based on previous individual communications with the nodes.
- We design and implement the unified MAC on the Linux kernel leveraging V-MAC as a base. Our real experiments show name-based communication latency can improve from 30s to 1.5s (20X) by leveraging address based rate adaptation algorithm while reducing loss rates from 15–20% to 0% for one consumer. Our current prototype is also able to select the ideal rate for three consumers while retaining low loss rates (1–4%).

II. Background

In this section, we provide all necessary conceptual and technical details for both communication paradigms and for commercial commodity WiFi dongles.

1) 802.11 Address Based Communication Paradigm: Address based communication Medium Access Control (MAC) relies on MAC addresses to communicate between nodes using source (addr2) and destination (addr1) MAC addresses. Relying on MAC addresses for commands and sending messages to nodes based on their identity is ideal for address communication, because the node to control is known beforehand. The MAC addresses are shared when a node joins a network which is defined by the user. The network is a BSSID address retained within every frame transmission (addr3). Once a node joins or creates a network, it announces itself by sending a beacon periodically (every 1, 10, or 100ms) that contains its supported rates and Time Synchronization Timestamp (TSFT) which is leveraged to synchronize nodes within the network's internal clock. The internal clock was used in older standards to send data over DSS or FHSS. However, radios since 1999 leverage OFDM which does not need TSFT. Other use cases of TSFT have been for power-saving techniques.

Once a node announces itself, each node within the network creates a station data structure shared between the kernel, SOC, and PHY for it internally to send data. The station structure is used by current frame rate adaptation algorithms to find the ideal rate for the target node (e.g., Minstrel). Frame rate adaptation is necessary in wireless communication to retain high goodput and low loss rate by changing the transmission rate based on received feedback (i.e. acknowledgement frames). There are multiple algorithms that are already developed and expected to be developed based on the station concept (e.g., sleep synchronization, MU-MIMO algorithms).

Most edge wireless communication leverages ad hoc mode which allows nodes to communicate directly (peer to peer) without an access point leveraging an independent station concept. Ad hoc requires discovering the network/group to join or create based on the BSSID (addr3) then periodic announcement (beacon) for nodes within the network to know about each other. Such design causes multiple problems: *i)* the arbitrary network discovery and creation stage creates latency and overhead that is unnecessary, *ii)* by enabling nodes to select a group instead of a single station, communication is not fine grained, which can lead to memory overhead within dense environments since for each node within the group, a station structure must be made and retained based on nodes beacons, and *iii)* with every node sending a periodic announcement at the base rate (1 or 6Mbps) in a dense IoT networks, the system does not scale well.

2) Name Based Communication Paradigm: Named Data Networking (NDN) filters and routes data based on a *dataname* which describes the content of the packet/frame instead of the transmitter or the receiver. V-MAC [7] is a medium access control layer for data-centric communication which provides filtering based on *encodings* that are hashes of a packet's *dataname*. V-MAC identifies the packet by hashing the *dataname* into the encoding. V-MAC leverages a Lingering Encoding Table (LET) [7] which stores a consumer's interest encoding to filter against. When the received frame has an encoding, the node checks its LET for an entry that matches that encoding, and it forwards the frame to the upper layers if there is a match. The LET differs from NFD's Pending Interest Table (PIT) [6] because the LET matches one encoding corresponding to a dataname to multiple frames; PIT matches one packet to one dataname packet. V-MAC also provides DACK, a robustness protocol that requests

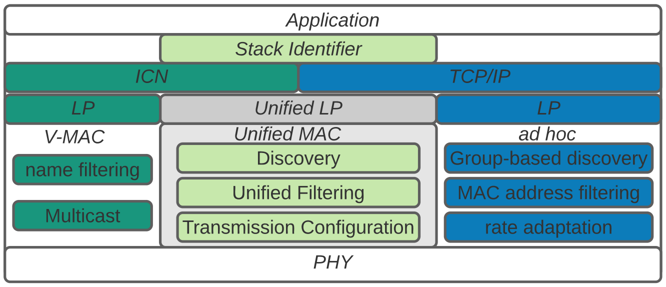


Fig. 1: Unified address and name based stack with the ability to do name based and address based filtering, and co-paradigm discovery protocol.

retransmission among neighbors on the fly without prior knowledge of the number of neighbors; it is not a rate control algorithm.

The existing V-MAC design does not support address-based communication, reliable one-hop one producer to one consumer communication, nor provide frame rate adaptation; in address based communication, algorithms providing these abilities have already been designed. Further, V-MAC is ideal for data dissemination and requesting distributed data without knowing the producer. However, it does not handle sending commands well by design of being data-centric.

Thus, edge device communication has two use cases: *i)* leveraging access control (command driven) and *ii)* sharing data (data driven). It is beneficial to combine both stacks to benefit from each paradigm's algorithms and advantages.

3) Commercial Commodity Wifi Dongles: Current commercial WiFi dongles have unicast frame rate adaptation, and most of the MAC layer implementations are in proprietary chipsets with no public access to the firmware. Even with chipsets that have public SOC firmware (e.g., ath9k_htc for 802.11n Atheros Qualcomm chipset AR9271), the station structure is at the core of the address based design. The station is used for frame rate adaptation, retransmission chains, and to provide statistical feedback regarding the target node to the upper layers. This station-based concept does not exist in data-centric communication, and thus prevents name based communication from gaining these benefits. Without redesigning the current 802.11 PHY, name based communication needs to leverage the address based architecture.

4) Related Work: Existing work has shown that content-based networks perform better than MQTT, a pub/sub protocol with a broker, due to elimination of the broker [10].We find that ample amounts of NDN work can benefit directly from our unified design. As an example, NAC [11], which supports data confidentiality and access control in NDN, needs to leverage high bandwidth data transmission on edge (unicast and multicast). Many other works [12]–[14] that rely on wireless communication can benefit from a unified MAC design that combines both name-based and address-based paradigms to benefit from both paradigms concurrently. One of the issues faced in earlier works [14] is high medium bandwidth overhead because using ICN over IP and existing address based MAC stack consumes bandwidth due to legacy overhead. These are issues we will eliminate by redesigning our unified MAC.

III. Design

A. Goals and Assumptions.

1) Goals: The goals of our design are: *i)* to enable applications to send address-based packets (e.g., commands) and to subscribe/publish name-based packets concurrently (e.g., file dissemination); *ii)* to reuse address based algorithms and mechanisms to benefit name based communication, and vice versa. As an example, currently there is an address-based communication rate adaptation algorithm for a single station; building an adaptive component to leverage the algorithm for single consumer transmission will benefit name based communication. *iii)* to design protocols using both paradigms where each communication is needed. For example, a name based discovery protocol can find stations based on attributes, eliminating arbitrary groups and allowing for fine-grained discovery.

2) Assumptions: We make the following assumptions: *i)* Address-based communication has existing protocols and upcoming designs that name-based communication can leverage with minor modification or adaptation. *ii)* Wireless edge communication applications leverage data-centric communication by requesting data (data dissemination), regardless of who provides it, and address-based communication by sending commands to particular nodes. *iii)* In dense WiFi deployment (e.g., dense IoT sensor building), the number of nodes within WiFi medium range increases to hundreds if not thousands, and a node may need to communicate with nodes of different communication paradigms based on its applications needs.

B. Overview

We provide an overview of how the whole stack will operate and the application's experience. Figure 1 shows the whole stack's overview structure and which components are name-based, address-based, or both. The *Stack Identifier* is necessary to abstract the two communication paradigms for the application developer. It enables making the appropriate decision based on *predefined parameters* (e.g., command/data dissemination, unicast/multicast, robustness levels, small/large data). *ICN* and *TCP/IP* are name-based (e.g., NFD [6]) and address-based network layers, respectively. Each network layer retains its own robustness and synchronization protocols.

We modify our previous work on V-MAC [7] and address-based communication (ad-hoc) to design cross-paradigm components. We illustrate how v-mac and WiFi stacks are unified and complement each other by three exemplary components: *i) filtering, ii)* adaptive component *Transmission Configuration, and iii) Discovery*. *i)* In filtering we reuse V-MAC's Lingering Encoding Table (LET) [7] to support address-based communication filtering using the same pub/sub mechanism. *ii)* The transmission configuration is for the name based paradigm to leverage existing features in the address based paradigm (e.g., frame rate adaptation to single node) and retaining node-pertinent information necessary in name based communication (e.g., link quality, supported rate, expected performance). *iii)* The discovery protocol eliminates address-based arbitrary grouping based on BSSID and allows discovering individual stations based on name-based subscription. It allows for more flexible and fine grained neighbor address discovery by reusing the same name based mechanisms.

Below we describe each of the components' design and changes we made to unify both communication paradigms. The goal is to provide one stack that enables both data-centric and address-based communication paradigms while benefiting from both paradigms' work when possible and co-operate to provide new features.

C. Filtering

The MAC layer analyzes incoming frames, and it filters by looking at frames' headers for either destination address, *encoding* value, or both. If the destination matches the node's local identity address(es), or the *encoding* value is in the LET (i.e. subscription sent), the frame is passed to upper layers; otherwise it is dropped. By enabling name-based and address-based paradigms to work concurrently, we enable receiving commands without subscription (address-based) and disseminating data upon sending an interest (name-based). We modify the LET to allow *Stable Entries* for addresses the node wishes to accept frames destined to (either node addresses, data names, or both).

Stable Entry. A node can have the need to filter based on multiple addresses (e.g., giving different nodes different virtual MAC addresses, or being a part of several multicast groups). Thus, we enable stable entries that store addresses the node wishes to allow receiving frames destined for (either indefinitely or a long time for multicast addresses). Our system also supports wildcard filtering for address-based communication. As an example, an entry with "b8:27:eb:*" will allow any frames with prefix matching "b8:27:eb" to be forwarded to the upper layers. These entries can be added and removed per upper layers' control. One additional benefit of having multiple reception addresses is to support monitor mode, where a node can receive the frames destined to any other nodes to examine the traffic. Thus the filtering of frames can be based on much more flexible means in both the number and format of addresses.

D. Transmission Configuration

Regardless of the communication paradigm used, every node has supported data rates, link quality, and other PHY parameters that are agnostic to filtering mechanism (address or name-based) and necessary in both communication paradigms. The *Transmission Configuration* packages all such information and allows for mapping it appropriately for both name and address based communication paradigms.

More specifically, the transmission configuration is used for unified communication paradigm frame rate adaptation. It stores *i)* supported rates by target node(s), and *ii)* the ideal observed rate (obtained from rate adaptation algorithms running e.g., Minstrel). For unicast, there are plenty of algorithms that can be leveraged to obtain node transmission configurations and to transmit name-based data frames using address-based algorithms. We offer a framework so different rate adaptation algorithms can be leveraged to operate on both communication paradigms.

Persistent Station. With stations being the core of address based communication, we reuse them for both address and name based communication. We retain stations discovered by our discovery protocol. An application may define a timeout duration (if a station has not responded to any frame within this time, it is assumed to be out of range or offline), or for an indefinite duration. A station can have a timeout or refresh duration in which a discovery frame is sent to check if the station is still available; otherwise the station entry is removed from the node's memory. The indefinite duration of the station entry works only if the node is designed to communicate with specific nodes by the application. Having nodes announce themselves (i.e. beacon) can cause heavy medium utilization [15] and thus we enable the response per discovery frame basis. An example scenario where both timeout options are used is of constrained sensors configured to only communicating with a control unit of a specific address; such nodes would set infinite timeout duration for the control unit station. However, the control unit may set a short timeout duration and rediscover the nodes to ensure their availability and fallback on backup sensors if needed. This also can be used as a notification mechanism for the administrators that such sensors are online within the network.

Mapping name based communication to address based communication. To leverage address based frame rate adaptation for name based communication, we need to identify the consumer interested in the data published. When a node sends an interest, it includes its source address. We include the consumer's source address as a private parameter while passing the request to upper layers. Thus, the ICN layer sends a name based data frame responding to the interest. We leverage the existing information about the target node from the interest to pass the frame rate adaptation algorithms the correct *station* to find the ideal rate for such station throughout the transmission and provide reliability. The transmission becomes address-based communication with rate adaptation only when one consumer is heard sending an interest for the data that a node is about to transmit. When multiple interests are received for the same data, multicast support is needed. We provide a list of all the *station* structures that asked for the data (i.e. sent an interest).

Multicast Transmission Configuration. Unlike unicast, there is no multicast rate control algorithm that exists. We leverage transmission configuration to select the initial data rate. We design a structure that leverages existing individual station entries obtained from address-based communication to construct a *group* structure for the supported rates and ideal rate. The policy followed creating the structure is to use the lowest data rate among all nodes transmitting to and supported rates by all nodes. If a node in the group does not have an existing station entry, the lowest data rate and minimum set of data rates is assumed by default. However, low-level ACK to adjust data rate optimally and further tune it (e.g., retransmission chain) cannot be supported as existing frame rate adaptations and frame-ACK only support unicast. Yet, finding the ideal rate leveraging recent history of transmission improves the goodput performance significantly over using base rate.

E. Discovery

The discovery protocol is designed with two goals: *i)* to eliminate medium utilization waste in address based station discovery by beacons, and *ii)* to support application level attributes to discover stations. The protocol enables discovering individual stations based on attributes the upper layers set. The attributes allow for subscription based on groups (e.g., *"temp"* for all temperature sensors) or more specific subscriptions (e.g., *"temp", "room", "A101" for temperature sensor in room A101.*

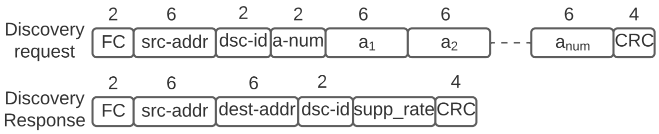


Fig. 2: Discovery protocol frame format. The discovery request frame contains Frame Control (FC) indicating type of frame, src-addr indicating the MAC address of the source node, dsc_id indicating the discovery identifier for the request, the number of attributes (a-num), and then attributes one by one (a). The discovery response has a discovery identifier, the supported rates with variable size (varies by 802.11 standard), and the destination address of the node of interest that matches the attributes in the request.

Attribute Table. Each node locally retains an attribute table that enables nodes to send *interest* frames with subtype discovery with one or more attributes. A node can have multiple attributes as handles to discover its identity (as stated earlier with temperature sensor having three attributes). The multi-attribute capability provides flexibility in describing the node per various attributes that can be linked to applications and services the node provides. One of the attributes can be the node's address, thus if any node sends a discovery for the particular node's address, it responds with its supported rates for the consumer node to build a station struct appropriately. Attributes can be used in multiple ways to discover nodes. It can be based on location (e.g., " Room A101"), functionality ("light control"), access credentials ("user: smith, pw: abc"), or combinations that can either be met independently or dependently.

Attribute Hash. Name based attributes can have variable lengths. We hash attributes to a fixed size to be placed in the frame header. As an example, for distributed edge computing, a node can add the attribute of "worker" and other more defining attributes. However, some of the attributes may have smaller or longer lengths. A hash length that is spatially (one-hop) and temporally (duration of the node using such attribute as a handle for its identity) collision free is needed. We empirically set the hash size to 6 bytes. We do not hash the MAC addresses as they are already 6 bytes in size and are unique by design.

Discovery frame. We create a new frame type for discovery protocol with two subtypes: request and response. We leverage a discovery identifier counter that maps discovery request attributes to a fixed size unique identifier that is local. The identifier is needed to associate a discovery response with a request without including all the attributes that were sent in the request. The reason we choose not to include the attributes and leverage a discovery frame identifier is because some frames may have up to 20 attributes with 6 bytes each, incurring much overhead. Thus, we leverage 2 bytes as the unique discovery identifier that is retained within the node locally to map the response MAC address to the node with the requested attributes.

Frame Format. Figure 2 shows the discovery frame format and response. Discovery request frames contain Frame control (FC) defining the discovery request type of a frame, the source address of the node locally sending, the discovery request identifier (disc_id) number of attributes, and the attributes (6 bytes each) with cyclic redundancy check (CRC) at the end. The discovery response frame consists of FC defining discovery response, the source address of the node being discovered, the destination address, the supported data rates by node, and the CRC. The supported data rates' size is variable, based on the standard and is defined by each 802.11 standard respectively. The frame format enables defining as many attributes to perform as fine-grained search as needed by the radio. The attributes to discover based on are passed from the running applications.

IV. Evaluation

A. Implementation

We build our system by modifying the V-MAC kernel module and further modify *ath9k_htc* firmware code [16] to support data rate indication from V-MAC and usage of its internal rate adaptation algorithm. We also further extend the *netlink* parameters passed from the userspace to a kernel module to indicate type of communication (data-centric or address-based) and include addresses when necessary.

We modify the filtering path by supporting checking the source MAC address and filtering against it. We also modify the transmission path to include the radio's MAC address in the frame as previously the source MAC address was spoofed in V-MAC (i.e. *ff:ff:ff:ff:ff*) hence it was not used. We evaluate our system's performance by using five nodes each consisting of a Raspberry Pi 4 and an Alfa AWUS036NHA WiFi dongle. We collect empirical data based on the five nodes' performance then do analysis to show how existing systems perform when scaling to hundreds and thousands of nodes nearby.

We benchmark our system's performance by analyzing the new unified paradigm's filtering mechanism, discovery protocol flexibility, and the benefit of leveraging address based algorithms for name based communication.

B. Filtering

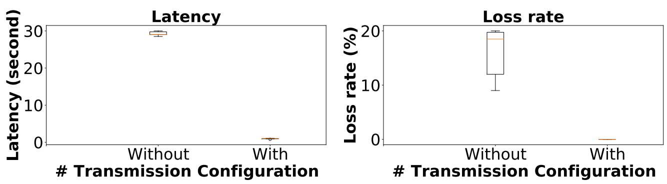
We validate our filtering capability by testing address and encoding combined. We test up to 3000 entries: 2990 name-based entries of interest (i.e. encoding), 5 addresses, and 5 addresses with wildcard (ignoring the last byte each). We find that the latency is constant for both complete addresses and encodings which is 20μs. However, with wildcard addresses, the filter has to iterate through one by one to see if a match exists, and that scales with the number of wildcard addresses. We find the latency increases by 20μs with each wildcard address added to the list (i.e. up to 100μs for filtering latency for five wildcard addresses). Generally, wildcard addresses should not have many entries as they encompass large sets of nodes based on prefix. Thus, we find that our design can support filtering against particular encodings and addresses efficiently.

C. Transmission Configuration

We evaluate our transmission configuration performance by sending five interests with broadcast address (i.e. how NFD operates currently) where each corresponds to 300 data frames (1400 bytes) back to back; then we repeat with our system to see the performance difference. We evaluate first with one producer and a single consumer, then with one producer and three consumers. Table I shows overall results of the experiments including medium utilization collected through Aletheia [15].

Number of Consumers	Latency	loss Rate	Medium
1 (broadcast)	30s	12 - 20% (median 19%)	35%
1 (TC)	0.9s	0%	1 - 3%
3 (Multicast)	2.16s	1 - 4%	2 - 5%

TABLE I: Overall Transmission configuration improvement results. The first row shows single consumer results without transmission configuration where data is transmitted as broadcast at base rate (1Mbps); the second row shows results of deploying transmission configuration (TC) that enables using address-based rate control algorithm and reliability to achieve lower loss rates, latency, and medium utilization. The third row shows multiple consumers which are known from prior communication performance through DACK and high data rate selection.



(a) Latency (b) Loss rate

Fig. 3: (a) shows the latency difference between using transmission configuration to map transmission from name-based encoding to station and broadcasting directly. (b) shows the loss rate difference between broadcasting at base rate (1Mbps) and using transmission configuration to achieve unicast.

Single consumer. Figure 3a shows the latency difference between using name-based communication directly and transmission configuration. Broadcasting data results in 30s latency and mapping it to the specific station that enables leveraging existing frame rate adaptation results in less than 1s latency. The 30X improvement comes from transmission using high data rates (65 Mbps) while broadcast defaults to interest and data transmission at 1Mbps due to unknown station information. Further, we can see that the loss rate of broadcast ranges from 5–30% while with transmission configuration, it is able to achieve 0% (Figure 3b). This is not a guaranteed reliability, but low-level reliability where a frame is retransmitted 3 times if no ACK is received. We also find that medium utilization for a single consumer drops from 35% to 1-3% by using transmission configuration. This is due to transmission configuration's ability to indicate that transmission is unicast, enabling existing address-based QoS (reliability), and rate control algorithm (leading to increase in data rate from 1 Mbps to 48 Mbps and 65Mbps) for the one consumer-producer scenario.

Multiple Consumers. Next, we have three consumers request the same data with prior communication with the producer as *persistent stations*. We merge the stations together (as described in Section III) and observe that the selected data rate was optimal giving 1–3% loss rate among all consumers with DACK protocol. The optimal rate selected in seven of the runs was 48Mbps and the rest was 54Mbps. However, without such aggregation, a higher data rate (65Mbps) results in higher loss rate (10–20%), and lower data rate reduces the goodput significantly. This shows that our aggregate structure of the stations of multiple consumers to find an optimal rate is better than randomly selecting a data rate or choosing the base rate (1Mbps).

We also test three consumers where one of them does not have any prior communication with the producer (i.e. the consumer does not have a station already for it). Our system defers to the base rate (1Mbps) since it the station link quality is unknown and it takes the most conservative approach, similar to broadcast. However, with our aggregate structure in place, finding the ideal rate through a probing algorithm becomes feasible.

D. Discovery

We evaluate our discovery protocol by performing analysis on the memory usage benefit gained from using our protocol over standard ad-hoc mode. We further analyze the overhead from doing the discovery based protocol over having every node send a beacon announcing itself from time to time.

Memory increases per stations needed to retain instead of nearby. We find that every station consumes about 60 bytes of memory. If we leverage ad-hoc, and have 200 nodes, the station structure alone will consume 12KB without all the other memory allocations needed. This becomes a fundamental issue when we go to higher number of nodes than 200 as stations structures must also reside within the PHY which has very small memory. Through our unified MAC, there is no need for such wasted memory. In our system, there is the need to only store stations that a node needs to communicate with based on application need.

V. Discussion

System Fidelity. Our system combines existing technologies of address- and name-based communication stacks in one MAC layer, complementing each other. We attempted to mitigate both systems' drawbacks by letting each stack do what it is ideal for (i.e. name-based data dissemination and discovery; address based commands and rate adaptation). We are investigating multicast frame rate adaptation as there is no algorithm for either the address based or name based paradigms. We envision more benefits to come from the unified design that both stacks cannot achieve solely. As an example, power saving in the address-based MAC layer currently relies on designating some other station to hold its information while it goes to sleep. By combining name and address based communication the correct station (address based) can be selected and the correct sleeping times can be selected based on application's data frequency (name based), improving system's performance. We also see more benefits to the name based communication paradigm by leveraging other address based communication algorithms (e.g., beamforming to multiple stations, MU-MIMO transmissions, AMPDU, etc.) We see our preliminary design as the first stage of name and address based hybrid design.

Nodes on different channels. 802.11 can run on different channels (three channels on 2.4GHz and three on 5GHz) with different bandwidths (20,40, 80, 160, 80+80 MHz). By eliminating networks and joining an arbitrary network, there are multiple channels that nodes can be distributed on. A protocol that establishes multi-channel station and data discovery is needed. With nodes free from group association in one channel, they can switch among different channels to request data, disseminate data, and send commands (e.g., a control unit obtains a video stream from one channel, disseminates data on another channel for processing, and sends commands to access control on third).

Discovery vs Beacon. WiFi leverages a beacon with the assumption that every node with WiFi turned on will need to be known all the time and announce its presence. In dense IoT networks, there are sensors that need to be available over wireless on demand but do not need to be known or announce their presence the whole time. As an example, a smart temperature and humidity sensor will not need to announce itself every 1ms as applications will only need their data every few hours as temperature and humidity do not change significantly in a short duration (unless extreme factors occur e.g., air conditioning turned on or windows opened). Further, these smart sensors can be placed in every room and multiple of them across the hallway. Having each sensor send a beacon announcing itself can cause heavy medium utilization overhead (up to 40% [15]) for no reason. With name based communication unified with address based communication, no node announcement is needed for such a scenario since a node can simply request the data itself directly. Thus, we believe for dense IoT networks and edge networks, the discovery option with application level timeout with minimal overhead is necessary to support hundreds and thousands of nodes for efficient medium utilization.

Tradeoff between rediscovery and storing stations. The overhead of rediscovery consumes medium utilization to rediscover a prior known station. However, it also provides a new indicator to the node rediscovering that such station is still nearby and available. If no rediscovery is done periodically, a node can potentially store many stations of nodes that are offline/shutdown. We believe associating the rediscovery and station storage to the applications running needs solves this problem (i.e. let the application discover when needed, and when the application terminates, remove its stations used).

Named-based paradigm usage Beyond Multi-Consumer Data Dissemination. The name-based paradigm can be used in other scenarios besides plurality of data: *i)* when the source/provider is unknown and the consumer only cares about the data; *ii)* in mobile scenarios where a consumer does not have the time to identify the nodes, and wants to fetch data quickly; *iii)* in dense IoT networks eliminating beacon overhead and per-neighbor state, and only retaining active per-subscription state.

Beyond 802.11. Our current work focuses on unifying the communication paradigms on 802.11 due to its wide usage in address-based communication paradigm and the rich assets that can benefit name based communication. Other radio technologies (e.g., LoRA, Zigbee, mmWave) can be unified in a similar manner to offer both address and name based features and functions.

VI. Conclusion

In this paper, we present a unified address and name based MAC that incorporates both stacks' strengths, providing a name based discovery protocol for address based communication, filtering based on name and addresses, and supporting single consumer name based rate adaptation through address-based algorithms with retransmission. Our experiments show single consumer latency improvement from 30s to 1s (30X) with loss rate reduction from 10–20% to 0%. The discovery protocol gives address based communication more flexibility by eliminating arbitrary grouping based on BSSID and enables discovering stations based on application requirements.

VII. Acknowledgment

This work is supported in part by NSF grants 1652276 and 1730291

References

[1] Nasir Abbas, Yan Zhang, Amir Taherkordi, and Tor Skeie. Mobile edge computing: A survey. *IEEE Internet of Things Journal*, 5(1):450–465, 2017.

[2] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017.

[3] Daisuke Nojima, Leonardo Lanante, Yuhei Nagao, Masayuki Kurosaki, and Hiroshi Ochi. Performance evaluation for multi-user MIMO IEEE 802.11 ac wireless LAN system. In *2012 14th International Conference on Advanced Communication Technology (ICACT)*, pages 804–808. IEEE, 2012.

[4] Dong Xia, Jonathan Hart, and Qiang Fu. Evaluation of the minstrel rate adaptation algorithm in IEEE 802.11 g WLANs. In *2013 IEEE International Conference on Communications (ICC)*, pages 2223–2228. IEEE, 2013.

[5] Shan-Hung Wu, Chung-Min Chen, and Ming-Syan Chen. Collaborative wakeup in clustered ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 29(8):1585–1594, 2011.

[6] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, KC Claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3):66–73, 2014.

[7] Mohammed Elbadry, Fan Ye, Peter Milder, and Yuanyuan Yang. Pub/sub in the air: A novel data-centric radio supporting robust multicast in edge environments. In *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 257–270. IEEE, 2020.

[8] A Rakesh Kumar. Smart network access for 802.11 based internet of things. In *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pages 315–320. IEEE, 2015.

[9] Juan Carlos Cano, Victor Berrios, Ben Garcia, and Chai K Toh. Evolution of IoT: an industry perspective. *IEEE Internet of Things Magazine*, 1(2):12–17, 2018.

[10] Frank T Johnsen, Lars Landmark, Mariann Hauge, Erlend Larsen, and Øivind Kure. Publish/subscribe versus a content-based approach for information dissemination. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 1–9. IEEE, 2018.

[11] Zhiyi Zhang, Yingdi Yu, Sanjeev Kaushik Ramani, Alex Afanasyev, and Lixia Zhang. NAC: automating access control via named data. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 626–633. IEEE, 2018.

[12] Jeff Burke, Alex Afanasyev, Tamer Refaei, and Lixia Zhang. Ndn impact on tactical application development. In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 640–646. IEEE, 2018.

[13] Tianxiang Li, Wentao Shang, Alex Afanasyev, Lan Wang, and Lixia Zhang. A brief introduction to NDN dataset synchronization (NDN sync). In *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pages 612–618. IEEE, 2018.

[14] Lorenzo Campioni, Mauro Tortonesi, Bastiaan Wissingh, Niranjan Suri, Mariann Hauge, and Lars Landmark. Experimental evaluation of named data networking (NDN) in tactical environments. In *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*, pages 43–48. IEEE, 2019.

[15] Mohammed Elbadry, Fan Ye, and Peter Milder. Aletheia: A lightweight tool for WiFi medium analysis on the edge. In *ICC 2021-IEEE International Conference on Communications*, pages 1–7. IEEE, 2021.

[16] *ath9k_htc*, 2017 (accessed August 10, 2021).