

The Hint Protocol: Using a Broadcast Method to Enable ID-Free Data Transmission for Dense IoT Devices

Yi Ren*, Ren-Jie Wu[†], and Yu-Chee Tseng*

*Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

[†]Garmin Ltd.

Abstract—IoT (Internet of Things) has attracted a lot of attention recently. IoT devices need to report their data or status to base stations at various frequencies. The IoT communications observed by a base station normally exhibit the following characteristics: (1) massively connected, (2) lightly loaded per packet, and (3) periodical or at least mostly predictable. The current design principals of communication networks, when applied to IoT scenarios, however, do not fit well to these requirements. For example, an IPv6 address is 128 bits, which is much longer than a 16-bit temperature report. Also, contending to send a small packet is not cost-effective. In this work, we propose a novel framework, which is slot-based, schedule-oriented, and identity-free for uploading IoT devices' data. We show that it fits very well for IoT applications. We propose two schemes, from an ideal one to a more practical one. The main idea is to bundle time slots with certain hashing functions of device IDs, thus significantly reducing transmission overheads, including device IDs and contention overheads.

Index Terms—Small data transmission, Communication Protocol, Internet of Things (IoT), Machine-Type-Communication, Wireless Network

I. INTRODUCTION

Internet of Things (IoT) traffic, which is characterized by *massive* connected devices and *small* data, introduces significant impacts on mobile network [1]. IoT devices range from small tags and sensors to more complicated actuators, mobile phones, and machines. Through these various things, we are able to closely connect the cyber world with the physical world. According to [2], the number of IoT devices will reach 18 billions in 2020. Such massively connected Internet devices may make significant impacts on mobile networks traffics [1]. IoT communications may rely on wireless networks, cellular networks, wireline networks, or even instant messaging. In this work, we focus on wireless access technologies. Statistics show that 50% of such packets are less than 100 bytes [3]–[5].

To support small data collection from a large number of IoT devices, the wireless network architecture should be carefully redesigned. In current wireless networks, e.g., Long-Term Evolution (LTE)/LTE-Advanced, the radio access part is designed for a rather low number of connections with continuous flows of information and relatively high data requirements. In IoT settings, a packet's ID and control signaling overhead is not negligible when its data payload is small. As an example, an IPv6 address is 128 bits, which is much longer than a 16-bit temperature report. *Can we solve the big header, small body problem by transmitting a packet without its ID?* In doing so, the communication overhead can be reduced significantly.

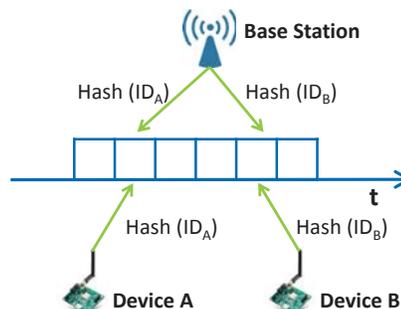


Fig. 1: Use a hash function as an agreement between two senders and a receiver.

Moreover, current cellular networks resource schedulers are designed to allocate resources on a per-user basis, which is very resource-consuming. For example, in LTE/LTE-A, a User Equipment (UE)-*specific* control signal is designed to schedule one UE at a time. This requires several dedicated control signals if an evolved NodeB (eNB) wants to send scheduling grants to several UEs. *Instead, can we schedule resources in a per-group manner? Can we broadcast a tiny hint signal so that UEs can decode the hint to get their grants scheduled by the eNB?* This can reduce signaling overhead significantly for massive connections.

Motivated by the above questions, we propose Hint, an ID-free data transmission protocol for IoT small data transmission. In the data transmission phase, we solve the big header, small body problem — a device can transmit its payload without attaching its ID in its packet. Also, the scheduler is able to allocate/schedule resources on a per-use, per-group manner. Our scheme can reduce the potential heavy contention among massive connections.

Allowing a device to transmit a packet without its ID included implies that its receiver knows where to collect the packet. To address this challenge, we borrow the idea of hash function to make an agreement between a sender and a receiver. Specifically, a common hash function is arranged between a base station and devices for mapping a key (e.g., ID) to an index in a table. In Fig. 1, devices A and B try to find their transmission opportunities and send their packets to the base station without ID included. To separate device A from device B, we utilize their personal information (e.g., ID) as input to the hash function. Considering the table as a time frame and the index as a time slot in the frame,

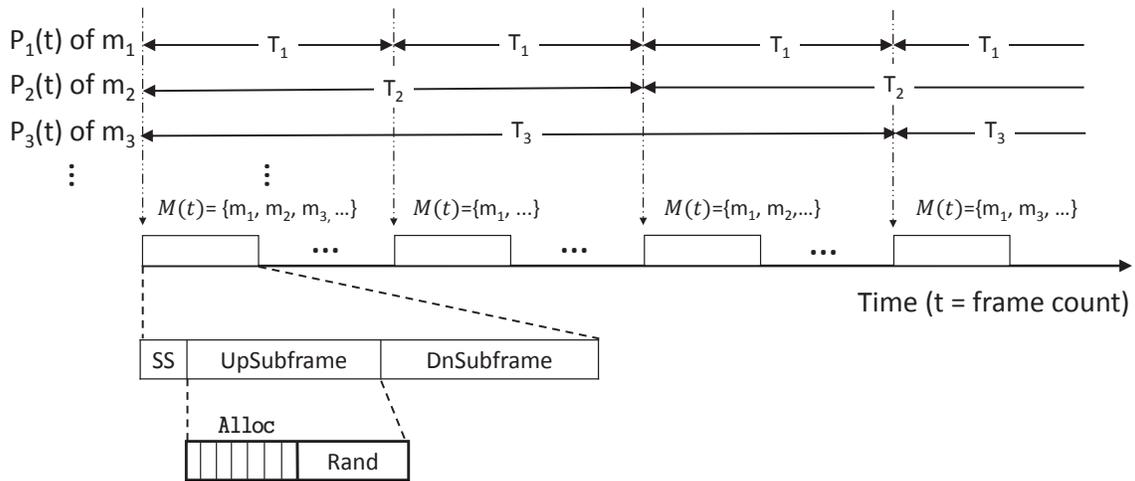


Fig. 2: The frame structure of our ID-free data collection schemes. T_1 , T_2 , and T_3 are transmission periods of m_1 , m_2 , and m_3 , respectively.

devices' transmissions can be mapped to time slots, which is a result of its hashing operations. By applying the same hashing operation, the base station can track who sent its data in which slot even if the sender's ID is not included.

With this basic idea in mind, here comes more challenges. What if two devices map to the same time slot? How to schedule devices without collisions problem? Given massive IoT devices, a broadcast-based control signal for scheduling grants on a per-group manner is preferred. The main idea is to bundle, in addition to device ID, a tiny *hint signal*, to the input of a *hint function* (e.g., hash function). Upon receiving the hint signal, a device can use (1) the hint signal, (2) a pre-determined *key*, and (3) a known hint function to extract information dedicated for its scheduling and/or configuration.

In this work, we design two versions of the Hint protocol based on the above framework. We show that channel utilization can be significantly improved, especially for small data transmission. For example, to transmit 10 bytes of payload, the Hint protocol can achieve up to 94% channel utilization compared with 38.4% in traditional scheme. Furthermore, our framework includes a mixture of contention and contention-free transmission modes. Both the contention level and transmission latency are significantly reduced.

The rest of the paper is organized as follows. Section II reviews the related work. Section III introduces our system model and problem statement. Section IV describes our framework and schemes, followed by our simulation results in Section V. Section VI concludes this paper.

II. RELATED WORK

Currently, the data collection approaches for IoT devices fall into two categories: (1) connection-oriented, and (2) connectionless. The connection-oriented approaches [6]–[8], for example, need to allocate channels or even time slots to UEs in advance. A connection should be established before any data transmission is possible, which is extremely costly for small

data transmission. Consider LTE/LTE-A as an example, a Radio Resource Control (RRC) connection establishment/release procedure includes more than 12 interactions in the Radio Access Network (RAN) side and 15 interactions in the Core Network (CN) side, no matter what the data size is. As a result, transmitting one bit of small data costs 5-70 times more signaling overhead compared to sending one bit of streaming data [9].

Alternatively, connectionless approaches (e.g., [9]–[12]) propose to skip the connection setup procedure for infrequent small data transmissions. In these approaches, devices transmit small data right after the random access procedure. That is, the small data traffic is piggy-backed with control messages. This means that the transmission happens in the control-plane and this may interfere control signals and thus incur longer latency for control signals. Recently, a piggyback mechanism is designed for NarrowBand IoT (NB-IoT) to transmit small data [12].

III. SYSTEM MODEL AND PROBLEM STATEMENT

We consider a set $M = \{m_1, m_2, \dots, m_d\}$ of d IoT devices covered by a base station, BS . Each IoT device needs to report its data or status to the BS from time to time. We consider the problem of collecting these devices' data and make the following assumptions:

- These devices are dense in the sense that d is much larger than that in typical Human-to-Human (H2H) communications.
- The data reported by each device in one transmission is small in the sense that it can be placed in one time slot. (For transmitting larger data, other mechanisms may be applied.)
- A device switches between two modes. When it has no intention to transmit data, it goes to the *non-connected* mode; otherwise, it switches to the *connected* mode.

- On entering the connected mode, a device m_i , $i = 1, \dots, d$, has to submit its *transmit pattern* $P_i(t)$ to the *BS*, where t is a frame counter maintained by the *BS* and all devices. $P_i(t) = 1$ if m_i intends to transmit in the t -th frames; otherwise, $P_i(t) = 0$. The simplest form of $P_i(t)$ is a periodical function. A more complicated one could be the combination of multiple periodical functions. From $P_i(t)$, the *BS* can derive whether m_i has data to transmit or not at frame t . (For unpredicted transmission needs beyond $P_i(t)$, there is a contention-based part to be used in our frame structure.)
- After entering the connected mode, a device has to maintain accurate time synchronization with the *BS*. However, under the non-connected mode, this is not needed so as to save energy.
- To support these IoT devices' data transmissions, the *BS* allocates a (logically) dedicated channel which contains a sequence of fixed-length *frames*. Each frame is divided into three parts: (1) Special part (SS): It is for the *BS* to broadcast important announcements to devices. (2) Allocation part (Alloc): It is divided into multiple slots for devices to transmit their data to the *BS* without carrying their IDs. (3) Random part (Rand): It is for any unscheduled/unpredicted transmission not arranged in Alloc and is used in a contention-based manner.

Fig. 2 shows the frame structure of our design. For any unexpected transmission or retransmission due to errors or collision, Rand can be used. We will propose several ID-free transmission schemes below. Note that the size of Alloc is adjusted dynamically, as will be clear later on.

IV. TWO ID-FREE TRANSMISSION PROTOCOLS

We propose two protocols, from an ideal one to a more practical one, for allocating slots to devices to transmit their data. The protocols have two nice features. First, to decrease the signaling cost, the *BS* will utilize broadcasting to announce only tiny control information to devices. Second, a device can transmit its data payload without attaching its device ID (such as IP or MAC address). Central to our protocols are (1) the hint signal, a tiny control information broadcast by the *BS*, (2) a pre-determined key, e.g., a device's ID, and (3) a known hint function, e.g., hash function, which can avoid the potential collisions among devices' transmissions as much as possible. To compute these hash parameters, the *BS* needs sufficient (but reasonable) computing power, as will be clear next.

A. Perfect Scheme (PS)

This scheme, as its name suggests, assumes an ideal hash function and can utilize the communication channel optimally. Let $h(ID, s)$ be a hash function, which takes a device ID and a seed s as inputs and generates an integer. We assume that function $h(\cdot)$ is pre-known by the *BS* and all connected devices when the protocol starts. Recall the transmit pattern $P_i(t)$ of m_i . At the t -th frame, the *BS* can compute the set of devices that intend to transmit:

$$M(t) = \{m_i | P_i(t) = 1, i = 1, \dots, d\}.$$

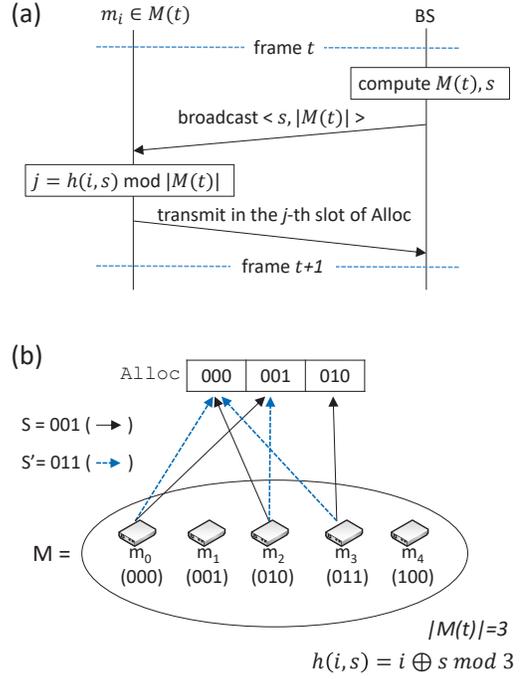


Fig. 3: (a) The message flow of PS. (b) An example of PS.

At the t -th frame, the protocol works as follows (refer to Fig. 3(a)):

1) The *BS* assigns $|M(t)|$ slots to Alloc and the rest of the slots to Rand. Then it tries to compute a perfect seed s , which maps each device $m_i \in M(t)$ to the time slot $h(i, s) \bmod |M(t)|$, such that

$$\forall m_i, m_j \in M(t), i \neq j : \\ h(i, s) \neq h(j, s) \pmod{|M(t)|}.$$

That is, there is no conflict in devices' hash results. Then the *BS* broadcasts $\langle s, |M(t)| \rangle$ as the hint signal in the SS to devices.

2) On receiving the hint signal $\langle s, |M(t)| \rangle$ in the SS, a device m_i with $P_i(t) = 1$ (intending to transmit) can upload its data to the *BS* in the $h(i, s) \pmod{|M(t)|}$ -th slot of Alloc.

The success transmission ratio λ_s of PS is 100%, where $\lambda_s = \frac{|\text{Alloc}|}{|M(t)|}$. In other words, PS enables all the intending-to-transmit devices (i.e., $M(t)$) to transmit in Alloc. Fig. 3(a) shows the message flow of PS. An example is shown in Fig. 3(b), where there are 5 devices, among which m_0 , m_2 , and m_3 intend to transmit in frame t . The hash function is designed as $i \oplus s$, where $i \oplus s$ is the exclusive-or operation. A proper seed is $s = 001$, which leads to a 1-to-1 mapping from m_0 , m_2 , and m_3 to slots 0, 1, and 2, respectively. On the contrary, $s' = 011$ is not a good seed because it leads to collision. Note that the scheme assumes the existence of a perfect function $h(\cdot)$ and the capability of the *BS* to find a perfect seed s , given any combination of $M(t)$. Therefore, the size of Alloc always equals $|M(t)|$ and its utilization

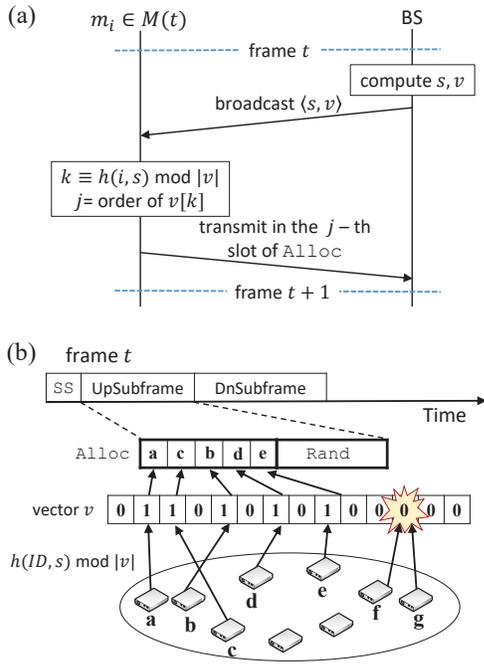


Fig. 4: (a) The message flow of VF. (b) An example of VF.

is always 100%. Only one broadcast message is needed to announce devices' slot allocation and the broadcast message size is $O(1)$.

We remark on some error-handling issues here. For a $m_i \in M(t)$ missing the broadcast in `SS`, it should not transmit in frame t and can retransmit in the `Rand` of a subsequent frame in which `SS` is correctly received (this is to tell the boundary between `Alloc` and `Rand`). If needed, the data in `Alloc` of frame t can be acknowledged in the subsequent subframe (`DnSubframe`). Similarly, if retransmissions are needed, it can use `Rand` of subsequent frames.

B. Virtual-Frame Scheme (VF)

This protocol tries to remove the requirement of finding a perfect seed in PS. It only guarantees a success transmission ratio $\lambda_s (\geq \lambda_{th})$ of intending-to-transmit devices to transmit in `Alloc` and needs a slightly larger broadcast cost in `SS`, where λ_{th} is a predefined threshold ratio such as 90%. The utilization of `Alloc` is still 100% through announcing a mapping vector called *virtual vector*, v , in `SS`. It works as follows (refer to Fig. 4(a)):

1) Define a binary vector v such that $|v| \geq |M(t)|$. The BS randomly picks up a seed s and computes the value of v as follows:

- Singleton case: Set $v[k] = 1$ iff there is exactly one $m_i \in M(t)$ such that $h(i, s) \bmod |v| = k$.
- Empty/Collision case: Set $v[k] = 0$, otherwise.

2) If the number of '1's in v is less than $\lambda_{th} \cdot |M(t)|$, go back to Step 1 to find another pair of $\langle s, v \rangle$. Otherwise, the BS chooses this pair $\langle s, v \rangle$ and broadcasts $\langle s, v \rangle$ as the hint signal in `SS`. Here, although the length of `Alloc` is not announced

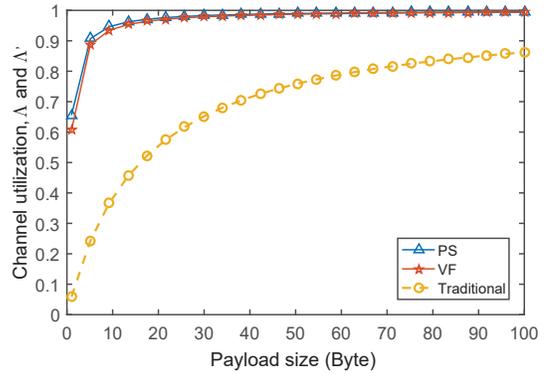


Fig. 5: Channel utilization vs. payload size.

explicitly, it is implied by the number of '1's in v . Also, note that $|v|$ is used for modular arithmetics.

3) Upon receiving the hint $\langle s, v \rangle$ in `SS`, a device m_i with $P_i(t) = 1$ can transmit its data in two ways. Let $k \equiv h(i, s) \bmod |v|$. If m_i finds $v[k] = 1$ (i.e., singleton), it can transmit in `Alloc[j]`, where j ($j \geq 0$) is the order of $v[k]$ in vector v , where the order of a bit in a binary vector is the number of '1's before it in the vector. If $v[k] = 0$, m_i has to contend for transmission in `Rand` of `UpSubframe`.

Note that when $|v| = |M(t)|$ and $\lambda_{th} = 100\%$, VF degenerates to PS. In practice, one can set a larger $|v|$ for accommodating more singleton cases in Step 1. The message flow of VF is shown in Fig. 4(a). Fig. 4(b) shows an example of VF, where there are 7 devices intending to transmit and a threshold $\lambda_{th} = 70\%$ is set. Let the length of v be 14. The figure shows a hash result where 5 devices find a singleton and 2 devices find a collision. So the length of `Alloc` is 5 and the success transmission ratio $\lambda_s = 71.4\% (> \lambda_{th})$. Device `a` can transmit in slot 0 because its hash result $v[1] = 1$ and there is no transmitter before it. Device `b` can transmit in slot 2 because its hash result $v[4] = 1$ and these are two transmitters before it. Devices `f` and `g` cannot transmit because they collide at `v[10]`.

Clearly, the scheme imposes a weaker requirement on $h(\cdot)$ due to a relaxed threshold ratio $\lambda_{th} (\leq 100\%)$ and a larger vector v ($|v| \geq |M(t)|$). The utilization of `Alloc` remains as 100% due to the broadcasting of vector v . Also, the remaining devices (a ratio of $1 - \lambda_s$) will be forced to contend for transmission in `Rand`. As to the computation cost, it is relatively easier to find a satisfactory s than PS. (Note that the scheme can still work even if a seed s not satisfying the threshold ratio λ_{th} is applied, by which the computation cost can be bounded.)

V. PERFORMANCE EVALUATION

In this section, we evaluate the proposed schemes from several factors. Section V-A compares channel utilization. Section V-B studies the impacts of hashing function and the length of virtual vectors.

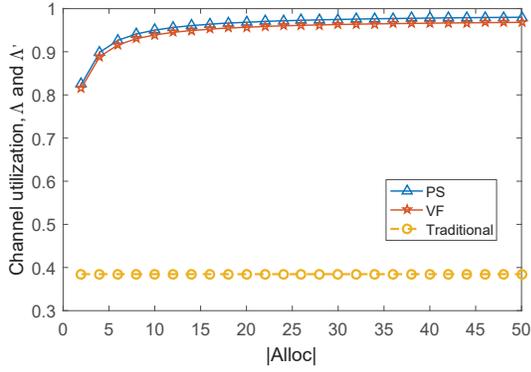


Fig. 6: Channel utilization vs. the number of devices allowed to transmit in $Alloc$.

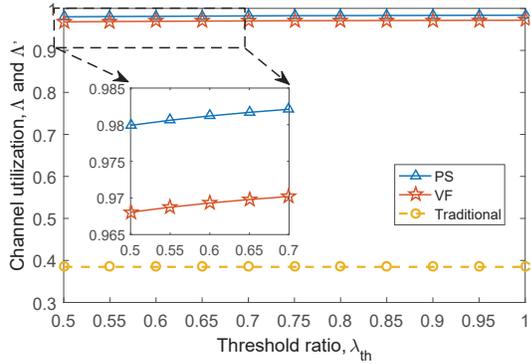


Fig. 7: Channel utilization vs. the threshold ratio λ_{th} .

A. Channel Utilization

The proposed schemes help reduce both packet header and medium access overheads. So they are especially suitable for IoT applications with small payloads. Here, we first ignore the access cost and evaluate the channel utilization of our schemes, denoted by Λ , which is defined as the payload divided by the payload and its overhead:

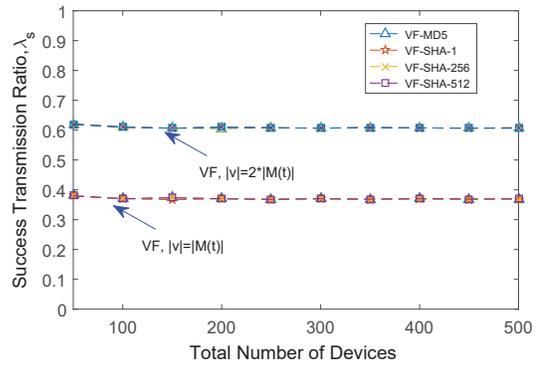
$$\begin{aligned}\Lambda &= \frac{\text{Size}(|Alloc|)}{\text{SS}(|Alloc|) + \text{Size}(|Alloc|)} \\ &= \frac{|Alloc| \times \text{payload}}{\text{SS}(|Alloc|) + |Alloc| \times \text{payload}}.\end{aligned}$$

Here $\text{SS}(|Alloc|)$ means the size of SS in anyone of our schemes when there are $|Alloc|$ devices allowed to transmit in $Alloc$. And $\text{Size}(|Alloc|)$ is the size of $|Alloc|$ slots. For traditional schemes, we define their channel utilization as:

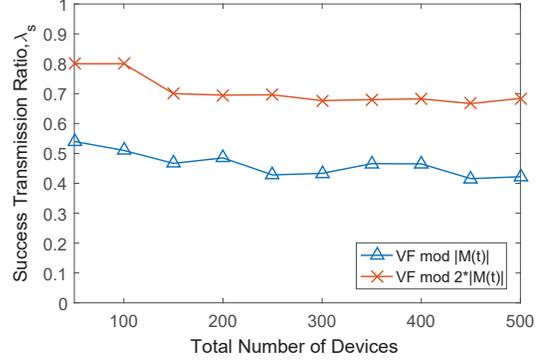
$$\Lambda' = \frac{\text{payload}}{\text{payload} + \text{header}}.$$

Next, we will evaluate the impacts of payload size $|Alloc|$ and threshold λ_{th} on Λ and Λ' . The seed size is set as 32 bits. The length of virtual vectors v_1, v_2, v_3, \dots are set to two times the remaining number of devices yet to be scheduled for transmission. For instance, the size of v_1 is 10 if there are 5 devices yet to be scheduled. The packet header is set to 128 bits (or 160 bits for IPv6).

1) *Impacts of payload size*: Fig. 5 compares the channel utilization of PS, VF, and traditional schemes by varying the



(a) Average value.



(b) Maximum value of SHA-1.

Fig. 8: Impact of hashing functions and the length of virtual vectors.

payload size. It is clear that the proposed schemes outperform the traditional scheme in this regard. PS has the highest Λ due to its strong assumption of finding a perfect seed. We also observe that the margins between proposed schemes and the traditional scheme are more significant when the payloads are smaller. This conforms with our goal of small data transmission for IoT devices.

2) *Impacts of $|Alloc|$* : Fig. 6 illustrates the utilization when varying the number $|Alloc|$ of devices allowed to transmit in $Alloc$. Clearly, a larger $|Alloc|$ would benefit our schemes. We observe that the performance of VF is very close to that of PS, which shows that VF is suitable for massive connected devices.

3) *Impacts of λ_{th}* : Fig. 7 depicts the comparison results by varying the threshold ratio λ_{th} , which is used to control the number of devices allowed to transmit in a round while computing a seed s . Its main purpose is to set an upper-bound for computation time. A bigger λ_{th} forces BS to compute a seed allowing more devices to transmit in a round. This reduces the use of virtual vectors, thus leading to smaller $\text{SS}(|Alloc|)$. In contrast, a smaller λ_{th} eases BS 's job in finding a satisfactory seed, causing less computation overhead. However, in terms of utilization, the impact of λ_{th} is not significant as shown in Fig. 7.

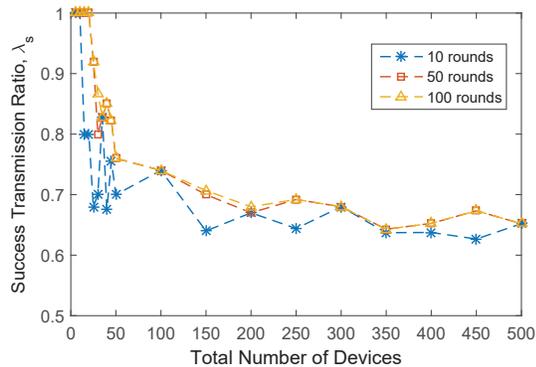


Fig. 9: Impact of the number of seeds being tried.

B. Impacts of hashing function $h(\cdot)$ and the length of $|v|$

Four hashing functions, MD5, SHA-1, SHA-256, and SHA-512, are evaluated. As above, we set the length of virtual vectors to be one or two times the number of devices yet to be scheduled in an iteration.

Fig. 8 depicts the impacts of these hashing functions and the length of virtual vectors. Specifically, Fig. 8(a) depicts the success transmission ratio, $\lambda_s = \frac{|\text{Alloc}|}{|M(t)|}$. We see that the impacts of these hash functions are almost negligible. Due to this result, in the following simulations, we only use SHA-1 for the rest of our performance evaluations. Here, we only try a small number of seeds in each iteration and use the best seed. In Fig. 8(b), the best seed is selected from 100 seeds. We observe that the best seeds enable 10% more devices to transmit in `Alloc` than average. Additionally, $|v|$ does have considerable impacts on the success transmission ratio λ_s . We can see that λ_s is about 0.4 with $|v| = |M(t)|$, whereas λ_s grows to 0.6 when $|v| = 2|M(t)|$. As $|v| = 2|M(t)|$ enables more devices to transmit in `Alloc`, the following simulations will set $|v| = 2|M(t)|$.

Fig. 9 further illustrates the impacts of the number of seeds that are tested in an iteration (here we tested 10, 50, and 100 seeds). We can see that one may find a perfect seed (i.e., $\lambda_s = 1$) from 100 random seeds when the number of devices is small (say around 20 devices in $|v| = 2|M(t)|$ case). Clearly, trying more seeds does help to increase λ_s . However, as the number of devices increases, the benefit declines. Based on our simulation results, trying 50 seeds is fair enough in most cases (which is also acceptable for modern base stations).

VI. CONCLUSIONS

This paper has proposed two protocols to the massive IoT transmission problem. Our protocols make the following contributions for small data transmission: (1) significant reduction in signaling overhead, and (2) elimination of ID when a transmission is conducted in contention-free mode, and (3) a mixture of contention and contention-free transmissions whose boundary can be precisely determined on-the-fly by the broadcast information by *BS*. We compare the proposed schemes against the traditional contention-based method and the results show that our proposed schemes can significantly reduce latency as well as increase resource utilization.

ACKNOWLEDGEMENT

This work is co-sponsored by MoE ATU Plan, MOST 105-2221-E-009-101-MY3, MOST 105-2221-E-009-100-MY3, MOST 105-2218-E-009-004, MOST 105-2218-E-009-008, MOST 104-3115-E-009-002, MOST 105-2218-E-009-003, Academia Sinica AS-102-TP-A06, ITRI, and Delta.

REFERENCES

- [1] Ericsson, *Ericsson mobility report, on the pulse of the networked society, Mobile World Congress Edition*, Feb. 2015.
- [2] CISCO, *Cisco visual networking index: Forecast and methodology, 2015–2020*, 2015.
- [3] R. Sinha, C. Papadopoulos, and J. Heidemann, “Internet packet size distributions: Some observations,” USC/Information Sciences Institute, Tech. Rep., May 2007.
- [4] W. John and S. Tafvelin, “Analysis of internet backbone traffic and header anomalies observed,” in *Proc. ACM SIGCOMM Conf. on Internet Measurement, IMC*, 2007, pp. 111–116.
- [5] 3GPP TR 37.869, *Study on Enhancements to Machine-Type Communications (MTC) and other Mobile Data Applications; Radio Access Network (RAN) aspects (Release 12)*, Std., Sep. 2013.
- [6] K.-D. Lee, S. Kim, and B. Yi, “Throughput comparison of random access methods for M2M service over LTE networks,” in *Proc. IEEE GLOBECOM*, 2011, pp. 373–377.
- [7] S.-Y. Lien, T.-H. Liau, C.-Y. Kao, and K.-C. Chen, “Cooperative access class barring for machine-to-machine communications,” *IEEE Trans. Wireless Commun.*, vol. 11, no. 1, pp. 27–32, 2012.
- [8] K. S. Ko, M. J. Kim, K. Y. Bae, D. K. Sung, J. H. Kim, and J. Y. Ahn, “A novel random access for fixed-location machine-to-machine communications in OFDMA based systems,” *IEEE Commun. Lett.*, vol. 16, no. 9, pp. 1428–1431, 2012.
- [9] Y. Chen, G. Li, Z. Pan, and I. Chih-Lin, “Small data optimized radio access network signaling/control design,” in *Proc. IEEE ICC*, 2014, pp. 49–54.
- [10] C.-H. Wei, R.-G. Cheng, and S.-L. Tsao, “Modeling and estimation of one-shot random access for finite-user multichannel slotted ALOHA systems,” *IEEE Commun. Lett.*, vol. 16, no. 8, pp. 1196–1199, 2012.
- [11] 3GPP TS 22.368, *Service requirements for Machine-Type Communications (MTC), (Release 13)*, 3GPP Std., Dec. 2014.
- [12] S.-M. Oh and J. Shin, “An efficient small data transmission scheme in the 3GPP NB-IoT system,” *IEEE Commun. Lett.*, Nov. 2016.