

Give Me a Hint: An ID-Free Small Data Transmission Protocol for Dense IoT Devices

Yi Ren, Ren-Jie Wu, Teng-Wei Huang, and Yu-Chee Tseng

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

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. 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. The framework is applicable from small-scale body-area (wearable) networks to large-scale massively connected IoT networks. Our simulation results verify that this framework is very effective for IoT small data uploading.

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

I. INTRODUCTION

Internet of Things (IoT) traffic, characterized by *massive* connected devices and *small* data, introduces significant impacts on mobile network traffic [1]. According to [2], the growth of the number of IoT devices may reach 50 billions in the next decade. Statistics show that 50% of IoT 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 cellular networks, the radio access part is designed for a rather low number connections with relatively high data requirements. In such settings, a packet's ID and control signaling overhead are particularly emphasized 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. *Is it possible to transmit the packet without its ID?* In doing so, the communication overhead can be reduced significantly.

Also, current cellular networks are connection-oriented (e.g., [6]–[8]); a connection should be established before any data transmission is possible, which is extremely costly for small data transmission. Consider Long-Term Evolution Advanced (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]–[11]) 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. Also, it violates the design principle of separating of user-plane and control-plane.

Moreover, a User Equipment (UE)-*specific* control signal is designed and transmitted to schedule one UE at a time, which is very resource-consuming. This requires several dedicated control signals if a base station wants to send scheduling grants to several UEs. *Is it possible to broadcast a hint signal so that the UEs with the hint are able to know their grants scheduled by the base station?* This can reduce signaling overhead significantly for massive connections.

In this paper, we propose an ID-free small data transmission protocol based on the IoT communication's characteristics: massively connected, lightly loaded per packet, and periodical or at least mostly predictable. Extended based on our recent results in [12], a two-virtual-frame (2VF) scheme is proposed. In our scheme, not only the random-access cost is largely eliminated, but also the signaling cost is minimized. The main idea of removing the random access cost is to bundle each time slot with certain hashing functions of devices' keys (e.g., ID). This allows us to directly tie a slot with a device's transmission opportunity without sending signals to devices individually. The signaling part is achieved by broadcasting a tiny *hint signal* to only those devices which have the intention to transmit in the upcoming transmission opportunities. 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. Moreover, because of our hint mechanism, in the data transmission phase, a device can transmit its payload without attaching its ID in its packet. Therefore, the communication costs are significantly reduced. We conduct through extensive simulations showing that our protocol significantly reduces latency as well as increases resource utilization. The results seem very promising for handling massive IoT communications.

The rest of the paper is organized as follows. Section II

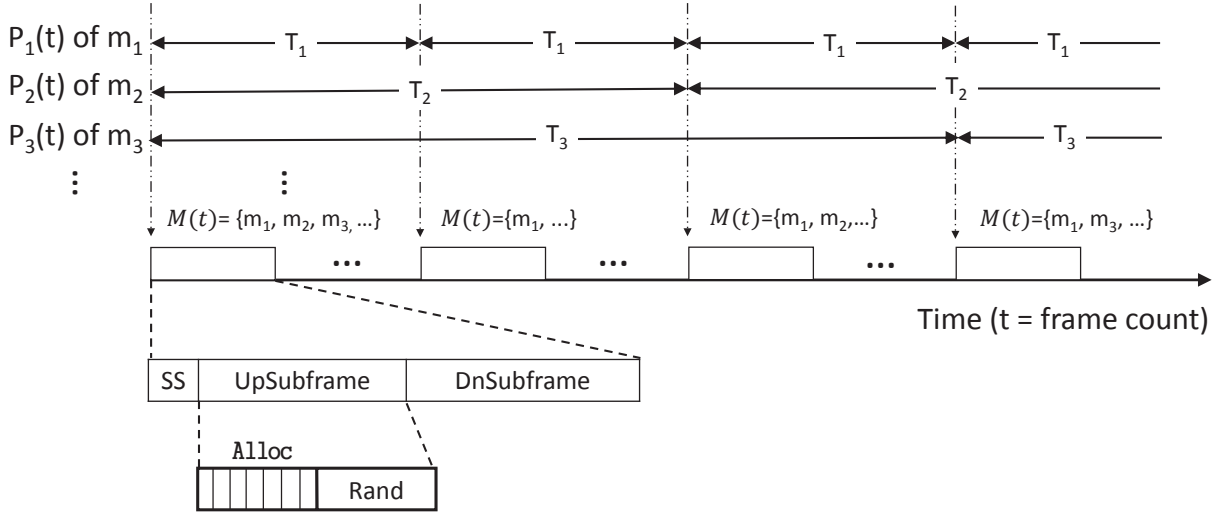


Figure 1: The frame structure of our ID-free data collection schemes. T_1 , T_2 , and T_3 are transmission periods of m_1 , m_2 , and m_3 , respectively.

introduces our system model and problem statement. Section III reviews our previous work. Section IV describes our framework and schemes, followed by our simulation results in Section V. Section VI concludes this paper.

II. 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. 1 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.

III. REVIEW: THE VF SCHEME

In [12], we proposed two hint protocols, perfect scheme (PS) and virtual-frame scheme (VF), 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 hint control information to devices. Second, a device can transmit its data payload without attaching its device ID (such as IP or MAC address). Here, we give a brief review of the VF. In particular, 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, hashing function, which can avoid the potential collisions among devices' transmissions as much as possible. To compute these hashing parameters, the BS needs sufficient (but reasonable) computing power, as will be clear next.

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:

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

Then it tries to compute a seed s , which guarantees a success transmission ratio $\lambda_s = \frac{|\text{Alloc}|}{|M(t)|}$ ($\geq \lambda_{th}$) of intending-to-transmit devices to transmit in *Alloc* where λ_{th} is a predefined threshold ratio such as 90%. The utilization of *Alloc* is 100% through announcing a mapping vector called *virtual vector*, v , in SS. At the t -th frame, the protocol works as follows (refer to Fig. 2(a)):

1) The *BS* assigns $|M(t)|$ slots to *Alloc*. 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 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 signal $\langle s, v \rangle$ in SS, a device m_i with $P_i(t) = 1$ can transmit its data in two ways. Let $k \equiv h(i, s) \pmod{|v|}$. 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*.

Fig. 2(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 hashing 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 hashing result $v[1] = 1$ and there is no transmitter before it. Device b can transmit in slot 2 because its hashing result $v[4] = 1$ and these are two transmitters before it. Devices f and g cannot transmit because they collide at $v[10]$.

IV. TWO-VIRTUAL-FRAME SCHEME (2VF)

VF forces a portion $(1 - \lambda_s)$ of devices to contend for transmissions in *Rand*. It is desirable to put as many devices into *Alloc* as possible. The 2VF scheme divides *Alloc* into two parts, *Alloc_1* and *Alloc_2*, and uses two seeds s_1 and s_2 to achieve this goal. There are two threshold ratios, λ_{th_1} and λ_{th_2} . *Alloc_1* is determined the same as the VF scheme, while *Alloc_2* tries to accommodate the remaining devices that cannot transmit in *Alloc_1*. Those devices that cannot be accommodated in *Alloc_2* have to transmit in *Rand*. It works as follows (refer to Fig. 3(a)).

1) The *BS* computes *Alloc_1* as follows. Similar to VF, a binary vector v_1 is defined such that $|v_1| \geq |M(t)|$. The *BS* repeatedly chooses a seed s_1 and computes v_1 as follows:

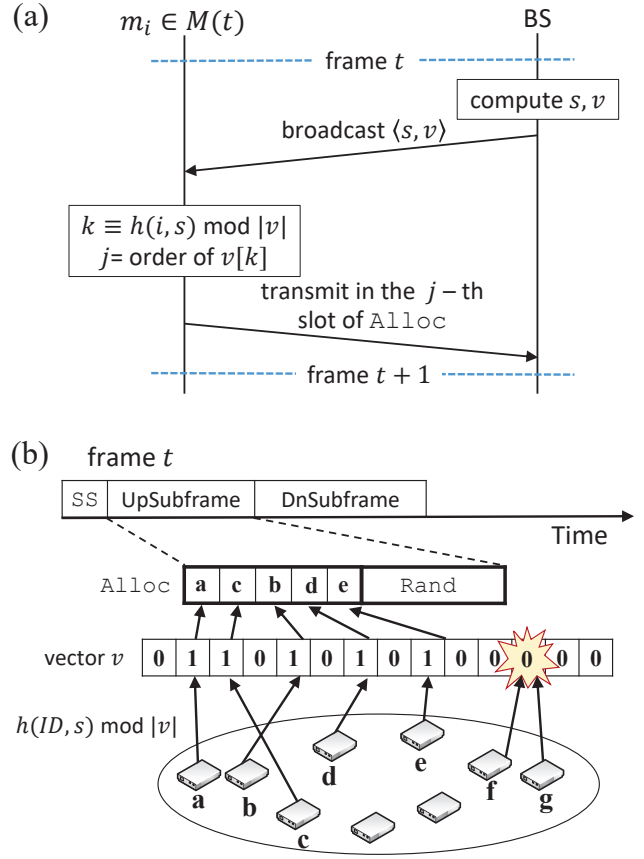


Figure 2: (a) The message flow of VF. (b) An example of VF.

- Singleton case: Set $v_1[k_1] = 1$ iff there is exactly one $m_i \in M(t)$ such that $h(i, s_1) \equiv k_1 \pmod{|v_1|}$.
- Empty/Collision case: Set $v_1[k_1] = 0$, otherwise.

The *BS* repeats the above trials and stops at an s_1 when the number of '1's in v_1 is larger than or equal to $\lambda_{th_1} \cdot |M(t)|$. The length of *Alloc_1* is the number of '1's in vector v_1 . The set of devices that can transmit in *Alloc_1* is

$$M_1 = \left\{ m_i \mid (m_i \in M(t)) \wedge (h(i, s_1) \bmod |v_1| = k_1) \wedge (v_1[k_1] = 1) \right\}.$$

2) Define a binary vector v_2 such that $|v_2| \geq |M(t) \setminus M_1|$. The *BS* repeatedly chooses a seed s_2 and computes v_2 as follows:

- Singleton case: Set $v_2[k_2] = 1$ iff there is exactly one $m_i \in M(t) \setminus M_1$ such that $h(i, s_2) \bmod |v_2| = k_2$.
- Empty/Collision case: Set $v_2[k_2] = 0$, otherwise.

The *BS* repeats the above trials and stops at an s_2 when the number of '1's in v_2 is larger than or equal to $\lambda_{th_2} \cdot |M(t) \setminus M_1|$. The length of *Alloc_2* is the number of '1's in v_2 . The set of devices that can transmit in *Alloc_2* is

$$M_2 = \left\{ m_i \mid (m_i \in M(t) \setminus M_1) \wedge (h(i, s_2) \bmod |v_2| = k_2) \wedge (v_2[k_2] = 1) \right\}.$$

The *BS* then broadcasts $\langle s_1, v_1, s_2, v_2 \rangle$ in SS.

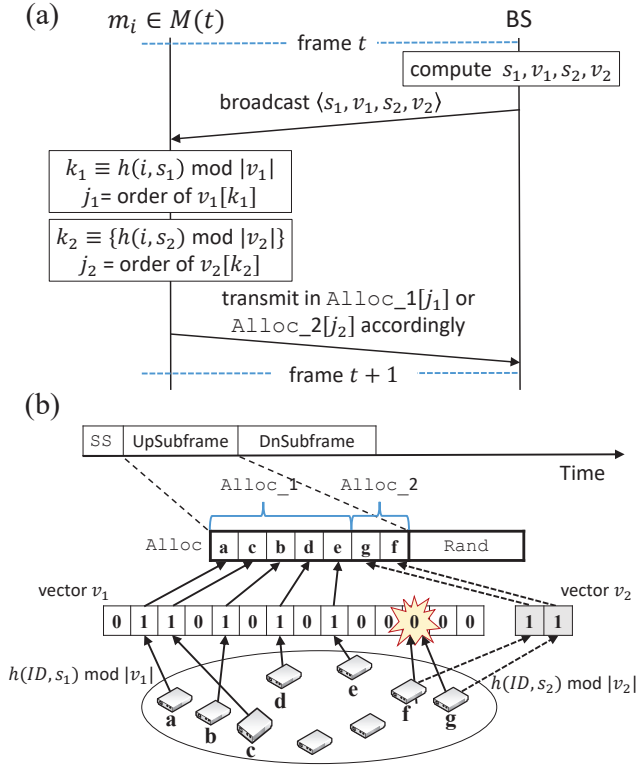


Figure 3: (a) The message flow of 2VF. (b) An example of 2VF.

3) Upon receiving the hint signal $\langle s_1, v_1, s_2, v_2 \rangle$ in SS, a device m_i with $P_i(t) = 1$ can transmit in three ways. Let $k_1 = h(i, s_1) \bmod |v_1|$ and $k_2 = h(i, s_2) \bmod |v_2|$.

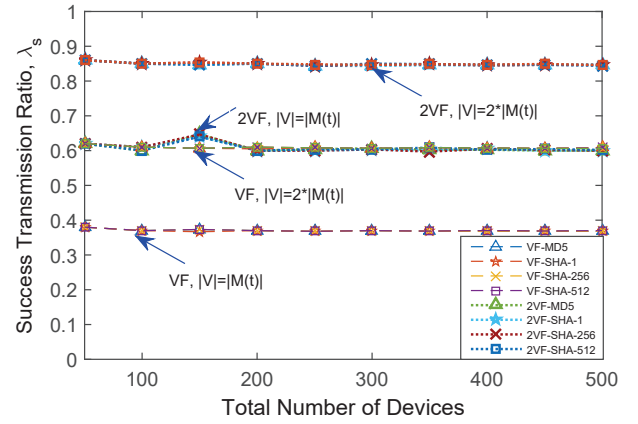
a) If $v_1[k_1] = 1$, it can transmit in $\text{Alloc}_1[j_1]$, where j is the *order* of $v_1[k_1]$ in vector v_1 .

b) If $v_1[k_1] = 0$ and $v_2[k_2] = 1$, it can transmit in $\text{Alloc}_2[j_2]$, where j_2 is the *order* of $v_2[k_2]$ in vector v_2 .

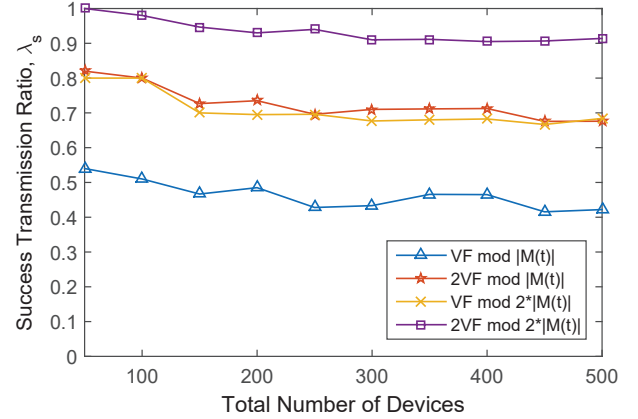
c) Otherwise, m_i has to contend for transmission in Rand of UpSubframe.

To summarize, 2VF runs VF twice with two vectors v_1 and v_2 using two thresholds λ_{th_1} and λ_{th_2} , respectively, so as to include more devices to transmit in Alloc. The message flow of 2VF is shown in Fig. 3(a). Fig. 3(b) shows an example using the scenario in Fig. 2(b) as the outcome of its first VF execution, so $|v_1| = 14$. There are two devices, f and g , remaining unable to transmit and it is quite easy to find a v_2 with $|v_2| = 2$ to schedule their transmissions in its second VF execution. Thus, no device needs to contend in Rand.

Our experience shows that even two reasonable ratios λ_{th_1} and λ_{th_2} can achieve a pretty high transmission ratio in Alloc. For example, by setting $\lambda_{th_1} = \lambda_{th_2} = 70\%$, at least $1 - (1 - \lambda_{th_1})(1 - \lambda_{th_2}) = 91\%$ devices can transmit in Alloc. By setting $\lambda_{th_1} = \lambda_{th_2} = 80\%$, the success transmission ratio can achieve 96%. In Section V-A, we will show that 2VF enables at least 80% devices to transmit in Alloc by performing only 2 times hashing operations with a virtual vector $|v_1| = 2|M(t)$, introducing ignorable computation overhead to a BS.



(a) Average value.



(b) Maximum value of SHA-1.

Figure 4: Impact of hashing functions and the length of virtual vectors.

V. PERFORMANCE EVALUATION

In this section, we evaluate the proposed schemes in terms of success transmission ratio and channel utilization.

A. Success Transmission Ratio

In the following, we first present the impacts of hashing function $h(\cdot)$, the length of vectors $|v|$, the number of seeds being tried on success transmission ratio λ_s .

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. 4 depicts the impacts of these hashing functions and the length of virtual vectors. We see from the figures that 2VF enables 20 – 40% more devices to transmit in Alloc than VF does.

More specifically, Fig. 4(a) depicts the success transmission ratio, $\lambda_s = \frac{|\text{Alloc}|}{|M(t)|}$. We see that the impacts of these hashing functions are almost negligible; the results of all four hashing functions almost converge in all settings. Due to the results, in the following simulations, we only use SHA-1 for the rest of our performance evaluations. To make a fair comparison, here we only try a small number of seeds in each iteration and use the best seed to set λ_s . The results are illustrated in Fig. 4(b),

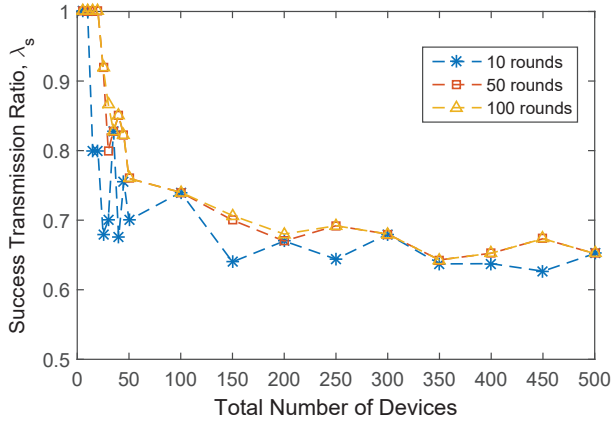


Figure 5: Impact of the number of seeds being tried.

where 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, the length of virtual vectors does have considerable impacts on the ratio λ_s . We can see that λ_s is about 0.4 with VF and $|v| = |M(t)|$, whereas λ_s grows to 0.6 when $|v| = 2|M(t)|$. Similarly, in the 2VF case, λ_s increases about 0.2 when $|v|$ grows from $|M(t)|$ to $2|M(t)|$. As $|v| = 2|M(t)|$ enables more devices to transmit in `Alloc`, the following simulations will set $|v| = 2|M(t)|$.

Fig. 5 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 are small (say around 20 devices in $|v| = 2|M(t)|$ case). Clearly, trying more seeds does help to increase the number of devices that can transmit in `Alloc`. However, as the number of devices increases, the benefit declines. Based on our simulation results, trying 50 seeds is fair enough considering λ_{th} and the computation cost (which is also acceptable for modern base stations).

B. Channel Utilization

Above, we have shown that 2VF outperforms VF significantly with respect to success transmission ratio. This reduces the number of devices contending via random access leading to less contention latency. In addition, 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 evaluate the channel utilization of our schemes, denoted by Λ , which is defined as the payload divided by the payload and its overhead:

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

Here $\text{SS}(|\text{Alloc}|)$ means the size of SS in our schemes when there are $|\text{Alloc}|$ devices allowed to transmit in `Alloc`. And $\text{Size}(|\text{Alloc}|)$ is the size of $|\text{Alloc}|$ slots. For traditional

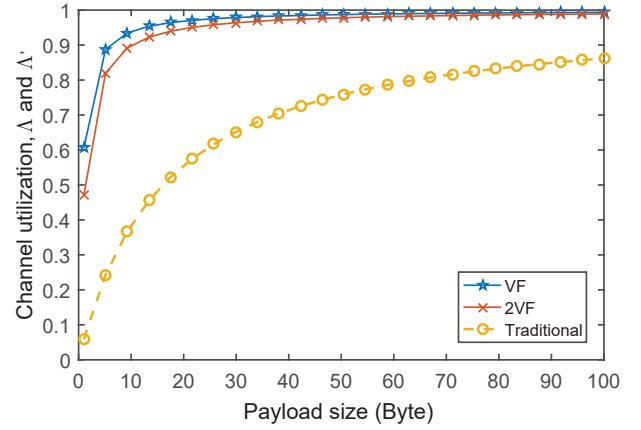


Figure 6: Channel utilization vs. payload size.

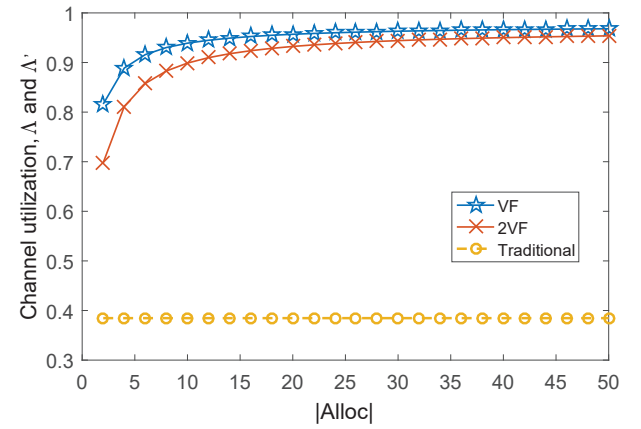


Figure 7: Channel utilization vs. the number of devices allowed to transmit in `Alloc`.

schemes, we define their channel utilization as:

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

Next, we will evaluate the impacts of payload size $|\text{Alloc}|$ and threshold λ_{th} on Λ and Λ' . The seed size is set as 32 bits. The length of virtual vectors v_1 and v_2 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. 6 compares the channel utilization of VF, 2VF, and traditional schemes by varying the payload size. It is clear that the proposed schemes outperform the traditional scheme in this regard. VF has the highest Λ . 2VF is worse than VF because $\text{SS}(|\text{Alloc}|)$ increases when two seeds and vectors are attached. However, we can see that the difference is very low. 2VF gains significantly on success transmission ratio with slight compromise on channel utilization. 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.

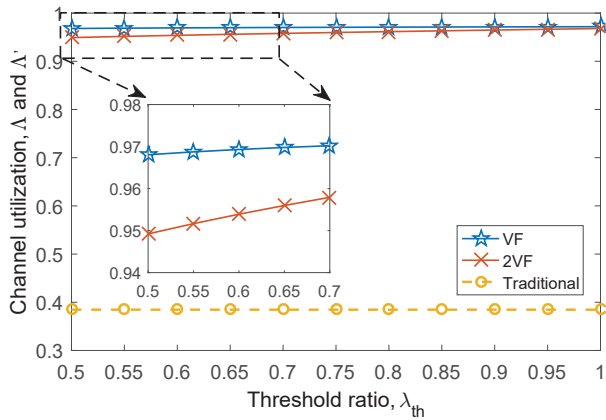


Figure 8: Channel utilization vs. the threshold λ_{th} .

2) *Impacts of $|Alloc|$* : Fig. 7 illustrates the utilization when varying the number $|Alloc|$ of devices allowed to transmit in $Alloc$. Clearly, a larger $|Alloc|$ would benefit our schemes. Since the value of $|Alloc|$ reflects the level of difficulty in finding a satisfactory seed for our schemes, we can see that a less number of iterations is preferred as $|Alloc|$ is smaller. However, as $|Alloc|$ increases, 2VF performs closely to VF. This shows that the proposed schemes are suitable for massive connected devices.

3) *Impacts of λ_{th}* : Fig. 8 depicts the comparison results by varying the threshold 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 $SS(|Alloc|)$. 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. 8.

VI. CONCLUSIONS

This paper has proposed a novel solution 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. The results seem very promising for handling massive IoT communications.

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] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *CISCO white paper*, Apr. 2011.
- [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. Liao, 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] Y. Ren, R.-J. Wu, and Y.-C. Tseng, "The Hint protocol: Using a broadcast method to enable ID-free data transmission for dense IoT devices," in *Proc. IEEE PerCom Workshop (PerIoT '17)*, Kona, Big Island, Hawaii, USA, Mar. 2017.