Postprint of a Group-based Multi-slot Parallel Identification Protocol for RFID Tags Compliant with the EPC Gen2 Standard

Authors: Xin' ai Yang, Duan Fu

Date: 2019-01-03T00:00:00+00:00

Abstract

Based on the analysis of the EPCglobal UHF Class 1 Generation 2 protocol and high-speed tag identification algorithms built upon the DFSA protocol, this paper proposes a Group Multi-Bit-slot Parallel Identification Protocol (GMBPIP) for RFID tags under the EPC Gen2 standard. This is achieved by adopting the flag slot setting method from the bit-slot FSA protocol for tag response and implementing a group bit-slot response flag word on the tags. The paper designs a novel group query command and a multi-group tag parallel identification protocol flow based on DFSA, theoretically analyzes the performance of the GMBPIP protocol, and conducts simulation experiments using the time parameters of the EPC Gen2 standard protocol. The results demonstrate that the GMBPIP protocol can effectively reduce the slot idle rate and collision rate under the EPC Gen2 standard while improving the tag identification rate and slot utilization, without imposing excessive computational burden on the tags. The average identification rate not only breaks through the 36.8% bottleneck of frame-slotted ALOHA protocols, but also surpasses the performance metrics of similar algorithms reported in current literature, achieving 70.95%~81.61%. GMBPIP can serve as a supporting protocol for high-speed identification of large quantities of passive tags in low-cost RFID systems.

Full Text

Preamble

Vol. 37 No. 3
Application Research of Computers
ChinaXiv Partner Journal

RFID Tag Group Multiple Bit-Slot Parallel Identification Protocol for EPC Gen2 Standard

Yang Xin' ai[1], Duan Fu[2]†

[1]Dept. of Computer Engineering, Shanxi Engineering Vocational College, Taiyuan 030009, China
[2]College of Information & Computer, Taiyuan University of Technology, Taiyuan 030024, China

Abstract: Based on analysis of the EPCglobal UHF Class-1 Generation-2 protocol and high-speed tag identification algorithms for DFSA protocols, this paper proposes a Group Multiple Bit-Slot Parallel Identification Protocol (GMBPIP) for RFID passive tags under the EPC Gen2 standard. GMBPIP employs the bit-slot response flag setting method from bit-slot FSA protocols by establishing a group bit-slot response flag word on each tag. The protocol introduces a new group query command and designs a multi-group parallel identification flow based on DFSA. Theoretical performance analysis of GMBPIP is presented, and simulation experiments were conducted using EPC Gen2 standard protocol timing parameters. Results demonstrate that GMBPIP effectively reduces slot idle rate and collision rate while improving tag identification rate and slot utilization ratio, without imposing excessive computational burden on tags. The average identification rate not only breaks through the 36.8% bottleneck of frame slotted ALOHA protocols but also surpasses performance metrics reported in current literature, achieving 70.95%~81.61%. GMBPIP can serve as a supporting protocol for low-cost RFID systems to identify large numbers of passive tags at high speed.

Keywords: EPC Gen2 standard; bit-slot FSA; passive tags; parallel identification protocol

0 Introduction

In recent years, Radio Frequency Identification (RFID) technology has been widely applied in supply chain management, warehouse management, and access control systems. Constrained by the RFID multiple access problem, one major challenge in such applications is efficiently reading large quantities of RFID tags attached to objects. Identification protocols play a crucial role in controlling multiple access and improving the efficiency of bulk tag identification. The Dynamic Frame Slotted ALOHA (DFSA) algorithm for passive tag identification, widely used in current RFID systems, is designed based on the EPC Gen2 (EPCglobal UHF Class-1 Generation 2) standard protocol [1].

In EPC Gen2, communication between readers and tags is organized using Frame Slotted ALOHA (FSA). The communication process is divided into frames, which are further partitioned into slots. The reader inventories tags within its coverage range slot by slot in frame cycles. Each slot results in one of three outcomes: an idle slot with no tag response, a successful slot with exactly one tag response, or a collision slot with multiple tag responses. Theoretical analysis shows that FSA achieves maximum channel efficiency when frame length L equals the number of unidentified tags, yielding tag identification, collision, and idle rates of 36.8%, 26.4%, and 36.8% respectively, with an average of 2.39 tags per collision slot. Consequently, estimating the number of unidentified tags and determining frame length accordingly becomes critical for achieving optimal channel utilization. In addition to classic methods such as minimum value, Poisson distribution, and collision factor approaches, sampling and composite estimation methods for large tag populations have emerged: literature [2] proposes a sampling-based linear tag estimation method; literature [3] presents a case-based tag estimation method using slot result sampling statistics; literature [4] introduces a slot count selection method that dynamically adjusts frame length based on collision slot count; and literature [5] summarizes ideal tag count ranges for different frame lengths and uses table lookup for dynamic frame length adjustment. However, due to estimation errors and EPC Gen2' s frame length determination method via Q value (Q [0,15], frame length L=2^Q), tag identification rate remains around 35%, deteriorating particularly when the number of unidentified tags is either small or large. To address this issue, literature [6,7] proposes an improved grouped DFSA algorithm that uses 354 (corresponding to frame length 256, Q=8) as the tag grouping threshold, dividing large tag populations into groups for sequential identification, and merging remaining tags from groups with few unidentified tags (tail tags) into subsequent groups.

To further improve identification rates for large numbers of passive RFID tags while reducing collision and idle rates, researchers have proposed various high-speed tag identification algorithms. These algorithms fully utilize non-successful slots to achieve identification efficiency above 45%. Literature [8] has tags generate two random numbers with priority levels, converting some idle slots into successful slots and raising identification efficiency to 45.3%. Literature [9] deploys three-level random numbers on individual tags, converting some idle and collision slots into successful slots to achieve a 69.35% identification rate. Literature [10] proposes a bit-slot grouped frame slotted algorithm that appends a bit-slot flag of length Ls to tag IDs. Tags randomly set one bit to 1 in this flag during response, with all other bits set to 0. The reader parses the position of the set bit and sequentially transmits position information back to tags from high to low. Tags in collision slots match this position information, and those that match successfully respond again. Theoretically, a single collision slot can identify up to Ls conflicting tags. Building on this, literature [11] proposes a segmented bit-slot FSA algorithm that divides bit-slots into two segments. When tag collisions occur, conflicting tags in the slot sequentially transmit the two segment flags for re-identification: if all conflicting tags in the slot are successfully identified in the first segment, the reader proceeds to the next slot; otherwise, the second segment is transmitted for further identification. This maintains an identification rate above 65% for tag populations under 10,000 with minimal communication overhead. Literature [12] combines grouping and multi-level random number concepts, deploying multi-level random numbers across different tag groups and proposes a group-based multi-level random number parallel identification algorithm based on the DFSA framework. This uses a polling mechanism for multi-level random numbers across groups and addresses load imbalance through inter-group merging, achieving 70% identification rate with 4 parallel groups. Literature [15] studies the problem of excessive query counts and low identification rates in existing algorithms, proposing a tag anti-collision algorithm based on multi-collision-bit detection.

While these algorithms improve tag identification rates, they demand high computational capabilities from tags and lack protocol support, requiring complex logic circuit implementations. This paper adopts the bit-slot response flag setting method from bit-slot FSA protocols and proposes a Group Multiple Bit-Slot Parallel Identification Protocol (GMBPIP) for RFID tags under the EPC Gen2 standard by establishing a group bit-slot response flag word on tags. Performance analysis is conducted for different numbers of parallel groups and bit-slots per group, with simulation experiments performed.

---

1 Parallel Identification Protocol Design

1.2 Command Definition

The proposed GMBPIP protocol is based on the EPC Gen2 standard. In addition to standard EPC Gen2 commands, it requires a group access control command **QueryRepG(gb, gp)**, where gb [1,m] with binary encoding length of $\log m$ bits, and gp [1,p] with binary encoding length of $\log p$ bits. The reader uses **QueryRepG(gb, gp)** to query the gb-th group of tags satisfying WGPS(gb, gp)=1. Tags in this group receiving **QueryRepG(gb, gp)** that match WGPS(gb, gp)=1 respond with their EPC. If no collision occurs, the tag is successfully identified; otherwise, it participates in the next round of identification. Table 1 lists the commands used in the protocol description.

Table 1 Commands for the GMBPIP Protocol

Command	Description
Select	EPC Gen2 command for selecting a specific tag set
Query(Q)	EPC Gen2 frame query command initiating a frame slot cycle, Q [0,15]
QueryAdjust	EPC Gen2 frame query command adjusting Q value relative to previous round (Q unchanged or ±1), initiating a frame slot cycle

Command	Description
QueryRep	EPC Gen2 slot repeat query command initiating the next slot
ACK	EPC Gen2 acknowledgment command for successfully identified tag EPCs
QueryRepG(gb,gp)	GMBPIP custom command querying the gb-th group of tags satisfying WGPS(gb,ps)=1, where gb [1,m], m [4,16] parallel groups; gp [1,p], p [1,4] multi-bit-slot count

1.3 Protocol Flow Design

The GMBPIP protocol proposed in this paper is based on the EPC Gen2 standard and enables parallel identification of multiple groups of passive tags. The detailed flow is described as follows:

GMBPIP Protocol Flow

Input: PGT={pgTj | j [1,m]}, Q [4,8] and m [4,16], with each tag having WGPS=0 and gb set.

Output: Identified tag set SPGT={spgTj | j [1,m]}, unread tag set UPGT={upgTj | j [1,m]}.

1. The reader estimates the tag set T within its coverage using the algorithm from literature [7] to determine tag count nt. Referencing the grouping method from literature [3], T is divided into ng groups suitable for frame length L 2^Q, Q [4,8], with 354 tags per group, yielding grouped tag set GT={gTi | i [1,ng]} where |gTi|=|T| and ng is an integer multiple of m [4,16]. The first m groups from GT are placed into PGT={pgTj | j [1,m]}. Following literature [10], each tag is configured with an m×p-bit binary group bit-slot response flag word WGPS, where p [1,4] is the bit-slot count per group response. WGPS is divided into m segments of p bits each, corresponding to response status of each tag group in a slot. WGPS(gb,1:p) denotes the response status of the gb-th group in a slot. When the first group responds, it randomly sets one bit to 1 among p bits in WGPS(1,1:p); when the second group responds, it randomly sets one bit to 1 among p bits in WGPS(2,1:p); and when the m-th group responds, it randomly sets one bit to 1 among p bits in WGPS(m,1:p).

2. Initialize the identified tag set for each group in PGT as SPGT={spgTj | j [1,m]}= , and the unread tag count set as UPGT={upgTj | j [1,m]}=|PGT|.

3. Tags with slot counter SC=0 randomly select one bit from the p bits in the WGPS segment corresponding to their group number gb, set it to 1, and immediately initiate response by transmitting WGPS.

4. The reader examines the returned WGPS bit-slot information. If no response signal exists in this slot, proceed to step 7.
5. The reader checks how many WGPS segments are non-zero. If n segments are non-zero, it indicates n groups responded in this slot. Set gb=1:
   - 5.1 Set gp=1. If the bit-slot information for segment gb is 0, proceed to step 6.
   - 5.2 For the bit-slot information of segment gb, if WGPS(gb,gp) 0, include this information in a group query command **QueryRepG(gb,gp)**. Tags in this slot receiving the command extract the position information and match it against their group number and corresponding WGPS bit-slot information. Successfully matched tags return their data packets to the reader. The reader detects returned ID information. If no collision occurs, it uses the **ACK** command for confirmation.
   - 5.3 The reader clears WGPS(gb,gp). If the bit-slot information for segment gb is not 0, increment gp and repeat step 5.2.
6. The reader clears the bit-slot information for segment gb. If WGPS 0, increment gb and repeat steps 5.1-6 until WGPS=0.
7. The reader sends a **QueryRep** command. Tags within range decrement their slot counter SC by 1 upon receiving it, then return to step 3. If this is the last slot in the frame, end the current round and return to step 1.
8. When all slots in a round are empty slots, the entire query process terminates.

In GMBPIP, the reader broadcasts **Query(Q)** or **QueryAdjust** commands to m tag groups, which generate random numbers SC between 0 and 2^Q-1 according to the Q value. For **QueryRep**, SC is decremented by 1. Within a slot, all tags with SC=0 transmit their group bit-slot response flag word WGPS. The reader statistically analyzes the bit setting pattern in each WGPS segment, then uses **QueryRepG** to query responsive groups by group and bit-slot. Tags in the queried group respond with their EPC encoding. If no collision is detected, a tag is successfully identified. The communication flow between the reader and tag groups in GMBPIP is illustrated in Figure 1 [Figure 1: see original paper].

2 Parallel Identification Protocol Analysis

The proposed protocol enables parallel identification of multiple tag groups, with each group containing roughly equal numbers of tags. The reader queries all groups slot by slot using identical frame slot cycles, while groups simultaneously respond via the group bit-slot response flag word WGPS. The reader analyzes the bit setting pattern in each WGPS segment, then uses **QueryRepG** to query responsive groups by group and bit-slot. Tags in the queried group respond with their EPC encoding, and if no collision occurs, a tag is successfully identified.

For any single group’s identification process, the fundamentals remain similar to single-group FSA protocols, with identical tag response rates PP1 (success rate PS1 + collision rate PC1) and idle rate PE1. The difference lies in the reader broadcasting group query commands **QueryRep** in each frame slot to obtain WGPS, adding L extra WGPS response slots per frame. For m processed groups, frame length L, and p bit-slots per group segment, the average number of non-zero WGPS group segments is SA=m×L×PP1, requiring SA additional slots using **QueryRepG** for group-by-group queries to ultimately identify tags. Since only responsive groups are re-queried, idle slots are converted into successful and collision slots. Group responses are reflected through p-bit segments, with values determined by responsive tags randomly selecting and setting one bit in the corresponding segment. With p bits per group segment compared to the original single bit, collision slots are reduced to 1/p of the original, converting some collision slots into successful slots. The probability of all bit-slots in a group being unresponsive is very low.

From this analysis, the identification rate PS, collision rate PC, and idle rate PE of GMBPIP are calculated using equations (1)-(3) when p=1, and equations (4)-(6) when p>1. Since single-group frame slotted ALOHA protocols achieve maximum tag identification, collision, and idle rates of 36.8%, 26.4%, and 36.8% respectively, with an average of 2.39 tags per collision slot, multi-bit-slot lengths of p=2,3,4 are appropriate.

Table 2 presents GMBPIP performance metrics for p=1 with m=1,4,8,16 calculated using equations (1)-(3). Table 3 shows performance for m=16 with p=1,2,3,4 calculated using equations (4)-(6).

Table 2 Performance Analysis of GMBPIP Protocol (One Bit-Slot p=1)

m	PS	PC	PE
1	36.8%	26.4%	36.8%
4	41.72%	29.93%	28.35%
8	48.61%	34.87%	16.52%
16	52.99%	38.01%	10.00%

Table 3 Performance Analysis of GMBPIP Protocol (Multi Bit-Slot p 1, m=16)

p	PS	PC	PE
1	52.99%	38.01%	10.00%
2	71.99%	19.01%	10.00%

p	PS	PC	PE
3	78.33%	12.67%	10.00%
4	81.5%	10.00%	10.00%

3 Simulation Experiments

Simulation programs for GMBPIP were developed in MATLAB using the timing parameters listed in Table 4 . Experiments were conducted for different parallel processing group counts and bit-slot counts per group. The timing parameters for GMBPIP simulation reference the EPC Gen2 standard protocol from literature [1] and literature [2]. In Table 4, Tari is the reference time interval for reader-to-tag signaling, with data-0 and data-1 durations of 12.5 s and 25 s respectively (80 kbps transmission rate). The tag-to-reader reverse link has data-0 and data-1 durations of 6.25 s each (160 kbps transmission rate). For the single-group case (m=1) without WGPS, the simulation program uses the improved grouped DFSA algorithm from literature [6,7].

Table 4 MPIP Protocol Timing Parameters

Parameter	Time (s)
TQuery	Query(Q) duration
TQueryRep	QueryRep duration
TQueryAdj	QueryAdjust duration
TQueryRepG	QueryRepG(gb,gp) duration
TWGPS	Tag WGPS response duration
TPEPC	Tag EPC response duration
TCollision	Tag collision response duration
TIdle	Tag idle response duration
TSuccess	Tag successful response duration

3.1 Simulation Experiments with Different Parallel Processing Group Counts

Simulations used passive tag populations ranging from 2,000 to 12,000 in increments of 500. Tag grouping estimation employed the same procedure as literature [13]. Parallel processing group counts were m=1,4,8,12,16, with results averaged over multiple simulation runs. Results are shown in Figures 2 [Figure 2: see original paper] through 4 [Figure 4: see original paper]. Figure 2 compares identification rate, collision rate, and idle rate. Figure 3 compares tag identification speed and tag read speed. Figure 4 shows curves for group count, frame (Q value) adjustment count, and multi-group unidentified tail tag merge count.

The results demonstrate that GMBPIP multi-group parallel protocols significantly outperform single-group protocols in tag identification rate and slot utilization (response rate), while frame (Q value) adjustment counts are substantially lower. Average identification rate breaks through the 36.8% FSA bottleneck, reaching 41.1%~57.4%, with tag identification speeds of 839~975 tags/s and read speeds of 448~485 tags/s. Slot utilization reaches 92%~97%. For GMBPIP multi-group parallel protocols, identification rate and slot utilization increase with parallel processing group count m, consistent with theoretical analysis. In some cases, multi-group tail tag merging enables the protocol to operate in the efficient identification range (256 tags per group), yielding performance metrics higher than theoretical values. However, improvement slows when m 12.

Table 5 compares GMBPIP multi-group parallel protocols against single-group protocols [11] for a total tag population of 147,000, evaluating identification efficiency, average identification speed, read speed, and runtime.

Table 5 Comparison of Simulation Results for Different Parallel Processing Groups in GMBPIP

Metric	m=1 (Single)	m=4	m=12	m=16
Average Identification Efficiency	35.8%	42.1%	55.05%	57.43%
Best Identification Efficiency	36.5%	51.14%	57.49%	59.72%
Worst Identification Efficiency	35.2%	41.4%	51.75%	52.58%
Average Identification Speed	803 tags/s	975 tags/s	920 tags/s	839 tags/s
Average Read Speed	438 tags/s	485 tags/s	448 tags/s	471 tags/s
Runtime	11.16s	9.55s	11.20s	9.88s

3.2 Simulation Experiments with Different Bit-Slot Counts

Simulations used passive tag populations from 2,000 to 12,000 in increments of 1,000. Tag grouping estimation employed the same procedure as literature [13]. Parallel processing group count was fixed at m=16, with results averaged over multiple runs. Results are shown in Figures 5 [Figure 5: see original paper] through 7 [Figure 7: see original paper]. Figure 5 compares identification rate and collision rate. Figure 6 [Figure 6: see original paper] compares tag identification speed and read speed. Figure 7 [Figure 7: see original paper] shows group count, frame (Q value) adjustment count, and multi-group unidentified tail tag merge count.

The results show that GMBPIP multi-bit-slot parallel protocols significantly outperform single-bit-slot protocols in identification rate and slot utilization, with lower frame (Q value) adjustment counts. Building upon the 41.1%~57.4% average identification rate of multi-group single-bit-slot protocols, multi-bit-slot settings further convert some collision slots into successful slots, achieving average identification rates of 70.95%~81.61%, identification speeds of 1,110~1,207 tags/s, and read speeds of 516~536 tags/s. These metrics exceed performance indicators reported in literature [9,11,12,14]. For GMBPIP multi-bit-slot protocols, identification rate and slot utilization increase with bit-slot count p, consistent with theoretical analysis. Performance may exceed theoretical values when multi-group tail tag merging enables operation in the efficient range (256 tags per group). However, due to the limit on tags per collision slot (average 2.39), improvement slows as p increases. The identification rate only increases by 3.6% when p grows from 3 to 4, making p=3 or p=4 most appropriate.

Table 6 compares GMBPIP multi-group parallel protocols against single-group protocols [11] for 77,000 total tags, evaluating identification efficiency, average identification speed, read speed, and runtime.

Table 6 Comparative Analysis of Multi Bit-Slot Simulation Results of GMBPIP Protocol (m=16)

Metric	p=1 (Single-bit)	p=2	p=3	p=4
Average Identification Efficiency	57.98%	70.95%	78.07%	81.61%
Best Identification Efficiency	60.41%	73.42%	80.01%	82.28%
Worst Identification Efficiency	55.22%	68.20%	76.05%	80.42%
Average Identification Speed	980 tags/s	1,110 tags/s	1,176 tags/s	1,207 tags/s
Average Read Speed	486 tags/s	516 tags/s	530 tags/s	536 tags/s
Runtime	2.41s	21.97s	19.61s	18.24s

4 Conclusion

Based on the EPC Gen2 protocol and considering the reader’ s strong computational capability versus the extremely limited resources of passive tags, this paper adopts the bit-slot response flag setting method from bit-slot FSA protocols. By establishing a group bit-slot response flag word on tags, we propose the GMBPIP protocol, design a new group query command, and develop a multi-group parallel identification flow based on DFSA. Theoretical performance analysis and simulation experiments using EPC Gen2 standard timing parameters demonstrate that GMBPIP effectively converts idle slots into successful and collision slots without imposing excessive computational burden on tags. Multi-bit-slot settings further convert some collision slots into successful slots, substantially reducing slot idle and collision rates while improving tag identification rate and slot utilization. The average identification rate not only breaks through the 36.8% DFSA bottleneck but also exceeds performance metrics reported in literature [9,11,12,14], achieving 70.95%~81.61% average identification rate, 1,110~1,207 tags/s identification speed, 516~536 tags/s read speed, and 92%~97% slot utilization. GMBPIP can serve as a supporting protocol for low-cost RFID systems to identify large numbers of passive tags at high speed.

References

[1] EPC Global. EPC radio-frequency identity protocols, class-1 generation-2 UHF RFID protocol for communications at 860 MHz~960 MHz, version 2.0.0 [S]. 2013.

[2] Chen W T. A feasible and easy-to-implement anti-collision algorithm for the EPCglobal UHF class-1 generation-2 RFID protocol [J]. IEEE Trans on Automation Science and Engineering, 2014, 11(2): 485-491.

[3] Duan Litian, Wang Zizhong J, Duan Fu. An identification algorithm in grouping and paralleling for data-intensive RFID systems [C]//Proc of the 1st International Conference on Big Data Computing and Communications. Cham: Springer, 2015.

[4] Daneshmand M, Wang C, Sohraby K. A new slot-count selection algorithm for RFID protocol [C]//Proc of the 2nd International Conference on Communications and Networking. 2007.

[5] Myung J, Lee W J, Srivastava J. Adaptive binary splitting for efficient RFID tag anti-collision [J]. IEEE Communications Letters, 2006(10).

[6] Lee S R, Joo S D, Lee C W. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification [C]//Proc of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services. Washington DC: IEEE Computer Society, 2005: 166–174.

[7] Wang Hui, Xiao Shengliang, Lin Feiyu. Group improved enhanced dynamic frame slotted ALOHA anti-collision algorithm [J]. Journal of Supercomputer, 2014, 69(3): 1235-1253.

[8] Shi Changqiong, Xiao Ruiqiang, Wu Dan. Improved anti-collision algorithm based dynamic frame timeslot ALOHA [J]. Computer Engineering and Design, 2014, 35(6): 1897-1900.

[9] Du Yongxing, Bai Wenhao, Li Baoshan. Improvement of the dynamic framed slotted ALOHA algorithm for RFID systems [J]. Chinese High Technology Letters, 2015, 25(3): 313-318.

[10] Li Hua. Research and performance analysis of UHF RFID multi-tag anti-collision algorithms [D]. Jinan: Shandong University, 2011.

[11] Xing Zhipeng, Yang Hengxin, Zhang Yun. A frame slot ALOHA algorithm of sectional bit-slot group [J]. Computer Technology and Development, 2016, 26(4): 31-35.

[12] Duan Litian, Wang Zizhong, Duan Fu. A parallel identification algorithm for RFID tags with group-based multiple-random numbers [J]. Journal of Taiyuan University of Technology, 2016, 47(4): 495-500.

[13] Duan L T, Zhang X Y, Wang Z Z, et al. Grouping-paralleling identification and authentication algorithm for RFID system EPCglobal G2V2 [J]. Journal of Computational and Theoretical Nanoscience, 2016, 13(5): 3183-3196.

[14] Zhao Lulu. Research on anti-collision algorithm of RFID tag [D]. Guangzhou: Guangdong Polytechnic Normal University, 2017.

[15] Fu Haolei, Lu Wangyan. Research of tag anti-collision algorithm based on multiple collision bits detection [J]. 2018, 35(12): 3757-3760, 3765.

Note: Figure translations are in progress. See original paper for figures.