International Journal of Engineering & Technology, 7 (2.8) (2018) 667-672

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

An approach to reduce turn around time and waiting time by the selection of round robin and shortest job first algorithm

Sarvesh Kumar[1*], Gaurav Kumar[2], Komal Jain[3], Aditi Jain[4]

[1](Assistant Professor), Computer Science & Engineering Department
JayotiVidyapeeth Women's University, Jaipur Rajasthan
[2](Assistant Professor), Department of ECSE, KLEF, Vaddeswaram,Guntur
[3](Research scholar), Computer Science & Engineering Department
JayotiVidyapeeth Women's University, Jaipur Rajasthan.
[4](Assistant Professor), Computer Science & Engineering Department, JayotiVidyapeeth Women's University, Jaipur Rajasthan.
**Corresponding Author Email: sarveshsingh@jvwu.ac.in*

Abstract

In this research, a study on operating system tells about its working, how it helps as interface between user software and system hardware .To implement this, different scheduling is used to provide multiple processing in a hardware. There are different levels of scheduler applied in different levels of process from ready queue to termination. This paper focuses on the average amount of waiting time and amount of turnaround time of processes. The proposed algorithm purely defines less waiting time and turnaround time as compared to the round robin scheduling and shortest job first scheduling algorithm.

Keywords: *Round Robin Scheduling, Shortest Job First, Scheduler, Operating System, Turnaround Time, Waiting Time*

1. Introduction

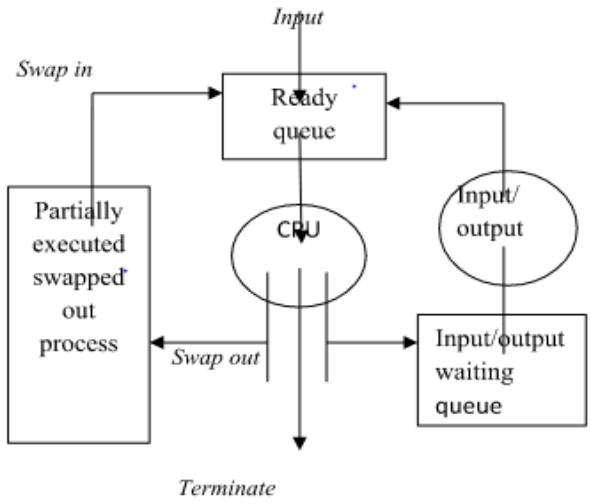
An operating system is a program which operates the system hardware. It helps in interfacing or connecting hardware to the software. It is designed in such a way which creates an interface between user and hardware.
This interface provides the user to interact with the hardware and execute programs. Operating System is responsible for simultaneous working of many users.
It is the most important part of system. Operating system is not only considered as windows, Linux, Mac .Any software which links or connects to the hardware is an operating system.
For example, if we consider a calculator there is no such operating software built in it but it has software programmed in C language which helps in interfacing to the calculator.
More summarized definition could be that the word operating system itself define as "operate" which means to access or to control the functioning and "system" which means any of hardware device,i.e. a substance or in language of computer science a software that access or control the functioning of a system.
We all know that the data is stored in the form of 0's and 1's in the hardware but the data could not be inserted into the hardware in form of 0's and 1's, here we need a software that is operating system which helps in converting our language into machine language. Whenever we input our data into a System using software, the Operating system helps it to convert the data in form of 0's and 1's.
There are no. of processes in a system i.e. when we start a program (program in execution is called process) it divides in no. of instruction those instructions are loaded in processor and finally output is generated .The processing of instructions are done through scheduling. There are some scheduling protocols which are applied before and after processor so that each process works smoothly.
To complete a program in execution we are required process scheduling. Before the systems were designed as a single processor i.e. they could perform only one task at a time, the rest of the processes need to wait for their chance to enter in CPU.
Nowadays the latest systems are multiprocessing. It does not mean that the no. of processors are added in the system, instead scheduling protocols are applied to execute program simultaneously.
Scheduling means something in a systematic manner. the same thing is done in our system .All the processes are scheduled according to some parameter. This helps in keeping CPU busy instead of sitting idle. The concept of multiprocessing was to maximize the CPU utilization.
When program is executed the no. of processes are stored in the memory .When a process has to wait for some reason like input/output request, the operating system handovers the CPU to another process, this saves time and increases CPU utilization.
There are three levels of process scheduling and these are described as:

2. Long Term Schedular

The long term scheduler opts to which processes are to be listed in the ready queue. The sequences of processes by which they will be executed in the processor are listed in the ready queue by long term scheduler. All the processes get executed after the scheduler selects the processes.

3. Mid Term Schedular

In midterm scheduler, the process is removed temporarily from the main memory to secondary memory (hard disk) or vice versa. This phenomenon is referred to as swapping of processes out. In swapping, if a process makes a request for input/output, the process is suspended and moved to secondary memory, this helps in the creation of space in the memory for another process.



4. Short Term Scheduler

This is also known as CPU scheduler. This scheduler fluctuate the state of process from ready queue to the running state. It picks a process from the ready queue which is ready to execute and allocate it to the CPU. The control over processes is done by short term scheduler i.e. it can even forcibly dispatch the process from a CPU and insert another process into CPU or it can keep executing the same process. This Scheduler also decides which process to execute next .This is faster than long term scheduler.

Five situations occur for the process from the ready queue to the termination.

1. The process in the ready queue switches from ready state to running state.
2. The process switches from running states to the waiting state when it request for input /output.
3. The process switches from the running states to the ready states.
4. When the process switches from waiting state to ready states.
5. When a process terminates.
   Admitted

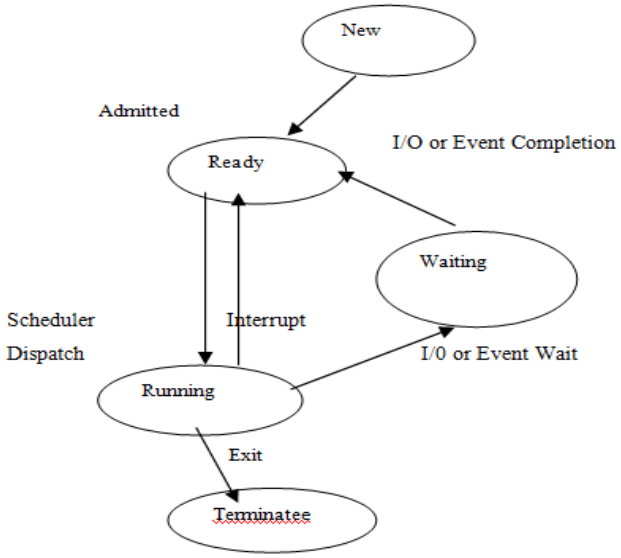


Fig. 2

5. Scheduling Criteria

There is some scheduling which have different properties from each other. Among these scheduling, according the situation the scheduling is decided. There are certain parameter which helps in distinguish, which scheduling must be used in particular situation.

CPU Utilization: In this, the CPU cannot be left idle. The CPU processing must be continuous i.e. it must remain busy. In real time system the CPU utilization reaches to 89.9% in hard real time system and 60% in soft real time system.

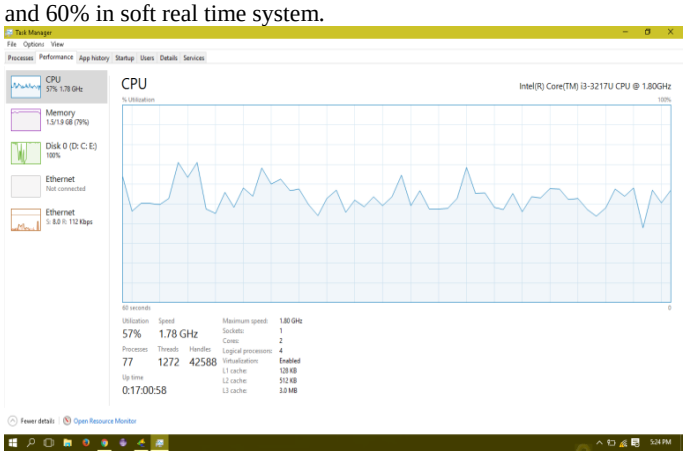


Fig. 3

This is a graph of CPU utilization of a normal user with operating system windows 10.

Throughput: Throughput stands for number of process executing per unit time. If the CPU is continuously busy than it will give a good throughput.

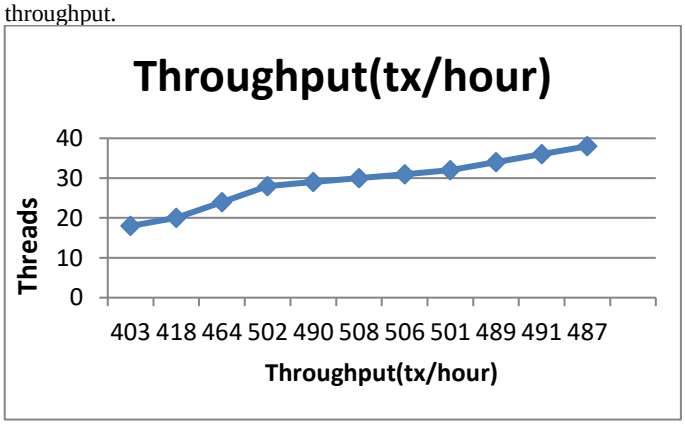


Fig. 4

Turnaround Time: The time spends by the process from the time of submission to the time of completion is turnaround time. Turnaround time is the complete time taken by the process to execute .It can be calculated as:

Turnaround Time = Completion Time-Arrival Time
i.e.

Process	Arrival Time	Burst Time
P1	0	5
P2	0	2

These two processes has arrival time same but different burst time so if we apply shortest job then we get different completion time for both the processes and we can calculate turnaround time.
So,

P2	P1

0 2 7

Here completion time for p1=7 and p2=2
So, turnaround time for p1:7-0=7
Turnaround time for p2:2-0=2

Waiting Time: The times spend by the process in a ready queue is the waiting time. It can be calculated as:

Waiting Time=Turnaround Time-Burst Time

Now we can calculate waiting time for both processes
Waiting time for process1
7-5=2
Waiting time for process2
2-2=0

The CPU scheduling does not acts on the amount of time during the process execution or does input/output, it only acts on the waiting time of process spend in a ready queue.

Response Time: It is the time when the process start responding for the 1st time, is called response time. It is not the time taken by the process to give output.

Context Switching: The switching of CPU from one process to another process is called context switching. The time used in switching must be less as it is wastage of time along with memory.

Now, we can optimize that a scheduling algorithm must pursue certain criteria for a real time and time sharing system.

CRITERIA	MINIMIZE	MAXIMIZE
CPU Utilization		✓
Throughput		✓
Turnaround Time	✓	
Waiting Time	✓	
Response Time	✓	
Context Switching	✓	

6. Simple Round Robin Scheduling Algorithm

The round robin scheduling algorithm can be defined as:-

1. A time quantum is set.
2. The processes are added to the ready queue.
3. The ready queue at as FIFO.
4. The 1st process is selected and executed up to the time quantum.
5. As the time quantum finishes two situation can be obtained:-
   - The burst time of process must have less or equal to the time quantum, then the process will release the CPU.
   - If the burst time of process is greater than the time quantum than the process is dispatched back to the ready queue.
6. Then the scheduler will pick up next process.

7. Simple Shortest Job First Scheduling Algorithm

The shortest job first scheduling algorithm can be defined as:

1. The process is loaded into ready queue.
2. The process with shortest next burst time is selected and executed .The process executes up to its burst time.
3. As the execution is completed the CPU selects for the process next with shortest CPU burst.

8. Proposed Scheduling Algorithm

1. A time quantum is set.
2. The process is loaded in to the ready queue.
3. The process with zero arrival time is selected and executed up to the quantum.
4. Two situation may occur
   - If the burst time is less than equal to time quantum than process is released.
   - If burst time is greater than the time quantum, the process is dispatched and added to the ready queue.
5. Then the process will be selected which have arrival time less than equal to the current starting time of next task i.e. if the CPU has time quantum 2ms and has reached to a time of

P1	P2

0 2 4

4 ms than the scheduler will select the processes with the arrival time less than equal to 4ms and among of that processes, the process with lowest burst time will be executed in CPU.
   - If the selected processes have same burst time, then select for the process with smallest arrival time among the same burst time process and if the arrival time is also same take the process first listed in the queue..
6. The process continues until the process reach to its completion time.

9. Case Studies

Let's take an example of five processes with defined CPU burst time and arrival time .These five process are scheduled in shortest job first, round robin, and proposed algorithm. This example shows the turnaround time, completion time, waiting time of each process and finally average time is calculated of turnaround time and waiting time .After calculation result of each scheduler is compared.

Experiment 1:-

This example consider five processes with its arrival time, burst time and with time quantum =2

PROCESS	ARRIVAL TIME	BURST TIME
P1	0	5
P2	1	2
P3	2	3
P4	3	4
P5	4	1

According to Shortest Job First Scheduling

Gantt chart

P1	P5	P2	P3	P4

0 5 6 8 11 15

Result:-

Process	Arrival time	Burst Time	Completion Time	Turnaround Time	Waiting time
P1	0	5	5	5	0
P2	1	2	8	7	5
P3	2	3	11	9	6
P4	3	4	15	12	8
P5	4	1	6	2	1
Total				35	20

Number of context Switch=4

Average Waiting Time=4ms

Average Turnaround Time=7ms

According to Round Robin Scheduling

Ready State:-P1 P2 P3 P1 P4 P5 P3 P1 P4

Gantt chart

P1	P2	P3	P1	P4	P5	P3	P1	P4

0 2 4 6 8 10 11 12 13 15

Result:-

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P1	0	5	13	13	8
P2	1	2	4	3	1
P3	2	3	12	10	7
P4	3	4	15	12	8
P5	4	1	11	7	6
Total				45	30

Number of context Switch=8

Average Waiting Time=6ms

Average Turnaround Time=9ms

According to proposed scheduling

Gantt chart

P1	P2	P5	P1	P1	P3	P3	P4	P4

0 2 4 5 7 8 10 11 13 15

Result:-

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P1	0	5	8	8	3
P2	1	2	4	3	1
P3	2	3	11	9	6
P4	3	4	15	12	8
P5	4	1	5	1	0
Total				33	18

Number of context Switch=5

Average Waiting Time=3.6ms

Average Turnaround Time=6.6ms

According to proposed scheduling

Gantt chart

P1	P2	P5	P1	P1	P3	P3	P4	P4

0 2 4 5 7 8 10 11 13 15

Result:-

An Approach to Reduce Turn Around time and Waiting Time by the Selection of Round Robin and Shortest Job First Algorithm

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P1	0	5	8	8	3
P2	1	2	4	3	1
P3	2	3	11	9	6
P4	3	4	15	12	8
P5	4	1	5	1	0
Total				33	18

Number of context Switch=5

Average Waiting Time=3.6ms

Average Turnaround Time=6.6ms

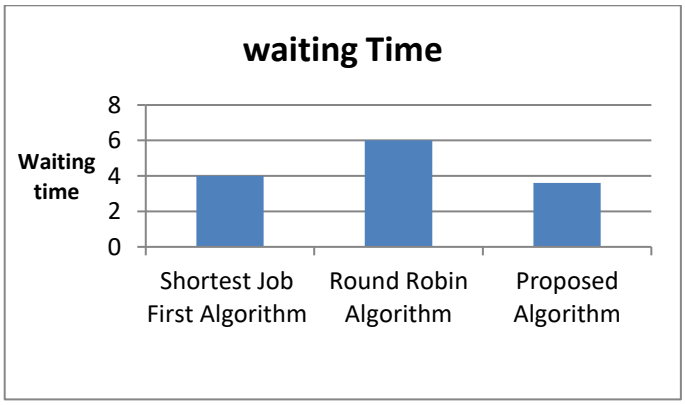


Fig. 5

Graph representation of three scheduling (Shortest job first, round robin, proposed) on waiting time.

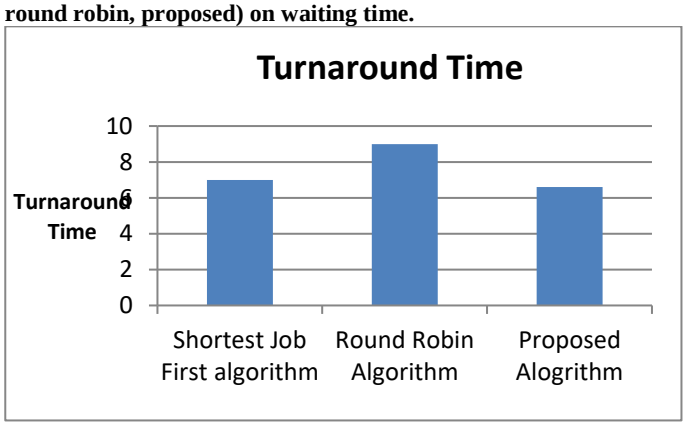


Fig. 6

Graph representation of three scheduling (Shortest job first, round robin, proposed) on turnaround time.

Experiment 2:-

This example contain four processes with its arrival time, burst time and with time quantum =2

PROCESS	ARRIVAL TIME	BURST TIME
P1	0	7
P2	1	6
P3	2	2
P4	3	5

According to Shortest Job First Scheduling

Gantt chart

P1	P3	P4	P2

0 7 9 14 20

To have non-visible rules on your frame, use the MSWord "Format" pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.

Result:-

Process	Arrival time	Burst Time	Completion Time	Turnaround Time	Waiting time
P1	0	7	7	7	0
P2	1	6	20	19	13
P3	2	2	9	7	5
P4	3	5	14	11	6
Total				44	24

Number of context Switch=3

Average Waiting Time=6ms

Average Turnaround Time=11ms

According to Round Robin Scheduling

Ready State:-P1 P2 P3 P1 P4 P2 P1 P4 P2 P1 P4

Gantt chart

P1	P2	P3	P1	P4	P2	P1	P4	P2	P1	P4

0 2 4 6 8 10 12 14 16 18 19 20

Result:-

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P1	0	7	19	19	12
P2	1	6	18	17	11
P3	2	2	6	4	2
P4	3	5	20	17	12
Total				57	37

Number of context Switch=10

Average Waiting Time=9.25ms

Average Turnaround Time=14.25ms

According to proposed scheduling

Gantt chart

P1	P3	P1	P1	P1	P4	P4	P4	P2	P2	P2

0 2 4 6 8 9 11 13 14 16 18 20

Result:-

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P1	0	7	9	9	2
P2	1	6	20	19	13
P3	2	2	4	2	0
P4	3	5	14	11	6
Total				41	21

Number of context Switch=4
Average Waiting Time=5.25ms
Average Turnaround Time=10.25ms

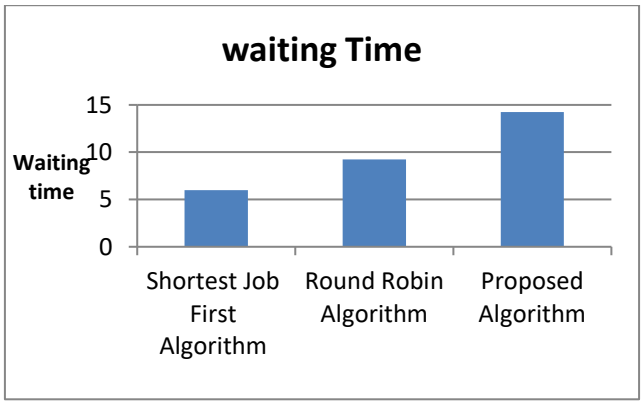


Fig. 7
Graph representation of three scheduling (Shortest job first, round robin, proposed) on waiting time.

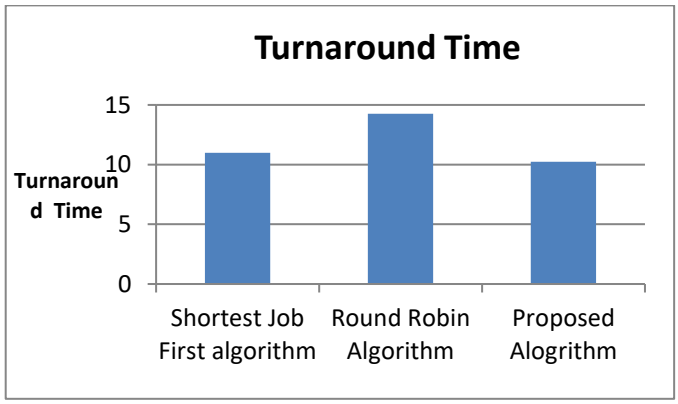


Fig. 8
Graph representation of three scheduling (Shortest job first, round robin, proposed) on turnaround time.

10. Conclusion

A comparative study is made on basis of round robin scheduling algorithm, shortest job first scheduling algorithm and proposed algorithm. The result obtained is that the proposed scheduling algorithm has less waiting time, turnaround time, context switching and less preemption as compared to round robin and less waiting time and turnaround time as compared to shortest job first. Therefore, it can be implemented on future work for real time system and time sharing system.

References

[1] Piotr D. Adamczyk and Brian P. Bailey. If not now, when?: The e ects of interruption at di erent moments within task execution. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '04, pages 271{278, Vienna, Austria, April 2004.

[2] Brian P. Bailey and Joseph A. Konstan. On the need for attention-aware systems: Measuring e ects of

[3] interruption on task performance, error rate, and a ective state. Computers in Human Behavior, 22(4):685{708, 2006.

[4] Mary Czerwinski, Eric Horvitz, and Susan Wilhite. A diary study of task switching and interruptions. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '04, pages 175{182, Vienna, Austria, April 2004.

[5] Dror G. Feitelson and Larry Rudolph. Gang scheduling performance bene ts for ne-grain synchronization. Journal of Parallel and Distributed Computing, 16(4), December 1992.Joel E. Fischer, Chris Greenhalgh, and Steve Benford. Investigating episodes of mobile phone activity as indicators of opportune moments to deliver noti cations. In Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services, pages 181{190, Stockholm, Sweden, August 2011.

[6] Conference on Human Factors in Computing Systems, CHI '05, pages 909{918, Portland, Oregon, April 2005.

[7] Eric Horvitz and Johnson Apacible. Learning and reasoning about interruption. In Proceedings of the 5th International Conference on Multimodal Interfaces, ICMI '03, pages 20{27, Vancouver, Canada, November 2003.

[8] Eric Horvitz, Andy Jacobs, and David Hovel. Attention-sensitive alerting. In Proceedings of the Fifteenth Conference on Uncertainty in Articial Intelligence, UAI'99, pages 305{313, Stockholm, Sweden, July 1999.

[9] Shamsi T. Iqbal and Brian P. Bailey. Understanding and developing models for detecting and di erentiating breakpoints during interactive tasks. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '07, pages 697{706, San Jose, California, April 2007.

[10] Shamsi T. Iqbal and Eric Horvitz. Disruption and recovery of computing tasks: Field study, analysis, and directions. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '07, pages 677{686, San Jose, California, April 2007.

[11] Shamsi T. Iqbal and Eric Horvitz. Noti cations and awareness: A eld study of alert usage and preferences. In Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, CSCW '10, pages 27{30, Savannah, Georgia, February 2010.

[12] Nickey Kern and BerntSchiele. Context-aware noti cation for wearable computing. In Proceedings of the 7th IEEE International Symposium on Wearable Computers, pages 223{230, Washington, DC, October 2003.

[13] N.D. Lane, E. Miluzzo, Hong Lu, D. Peebles,T. Choudhury, and AT. Campbell. A survey of mobile phone sensing. Communications Magazine, IEEE, 48(9):140{150, Sept 2010.

[14] Kyungmin Lee, Jason Flinn, T.J. Giuli, Brian Noble, and Christopher Peplin. AMC: Verifying user interface properties for vehicular applications. In Proceedings of the 11th International Conference on Mobile Systems, Applications and Services, pages 1{12, Taipei, Taiwan, June 2013.