ENHANCING SERVICE DISCOVERY PERFORMANCE OVER HOME NETWORKS

Intisar Al-Mejibli[1] and Martin Colley[2]

Department of Computer Science and Electronic Engineering, University of Essex, Colchester United Kingdom
[1]ialmej@essex.ac.uk
[2]martin@essex.ac.uk

ABSTRACT

Service discovery protocols are used to facilitate dynamic cooperation among devices / services with minimal user intervention. These protocols work by exchanging messages to identify and locate the relevant services. When new devices and services are added the flow of messages often appears as a burst. These bursts act as a critical factor that may cause the loss messages which dramatically influences the performance of the service discovery process. Thus, the loss of messages results in uncompleted discovery process which will directly affect the sharing of resources in home networks.

This paper proposes an algorithm that computes the minimum period of time required between a consecutive burst of messages and determines the minimum required queue sizes for the routers to manage the traffic and avoid dropped messages. The algorithm has been applied to the Universal Plug and Play (UPnP) protocol and considered the used of Active Queue Management (AQM). It was tested when the routers were connected in two configurations; decentralised and centralised. The message length and bandwidth of the links among the routers were taken into consideration. The result shows better improvement in number of dropped messages `among the routers.

KEYWORDS

Dropped messages, Service discovery protocols, Adaptive Queue Management (AQM).

1. INTRODUCTION

Home networks contain a wide range of networked devices in order to increase the quality of life, and empower users by providing information and services in an effortless way. Such devices and services will be highly interconnected, but usually cannot depend on pre-configured servers or static network addresses; instead these environments will typically be highly dynamic.

Home networks could consist of anything a homeowner can imagine, ranging from large domestic appliances such as the fridges, microwaves and audio-visual equipment to the lightweight temperature and smoke sensors. In addition mobile devices, smart cards, bar codes in grocery packages and 'smart' clothing and accessories must also be included. The main goal of interconnecting the home devices together is to share the network services and resources, and to invoke them remotely. Many protocols have been proposed to achieve this purpose which is to locate and invoke the services and resources in network, known as services discovery protocols [1]. Most of the service discovery protocols rely on the exchange of messages to locate remote services and to provide access to them. Sending too many messages into the network from multiple nodes at the same time could cause congestion which will lead to router queue overflow and the loss of messages. Accordingly, more messages must be sent to discover the services in the network and this causes more latency in discovery process and greedy

DOI : 10.5121/ijcnc.2012.4206

consumption of the network resources. The router queue management algorithm controls the queue strategy directly thus it should allow temporary bursty traffic, and penalize flows that persistently overuse bandwidth to avoid dropping packets. Many AQMs have been used to manage the queues, for example Drop Tail [2], Random Early Drop (RED) [3] and Random Exponential Marking (REM) [4] each of them has specific strategy.

This paper discusses how to avoid dropped messages during service discovery process in small networks which fall into the LAN (Local Area Network) category such as a networked office building, or home. In addition, it proposes an algorithm to overcome this significant issue, in order to make the discovery process perform smoothly and seamlessly. Further it explains the impact of the used AQM with the proposed algorithm on final results.

The structure of this paper is as follows. Section 2 introduces the related work which includes service discovery protocols, AQM protocols, and available mechanisms and algorithms that have been proposed to avoid or minimize the number of dropped messages. Section 3 introduces the proposed algorithm. The simulated model is detailed in section 4. Finally, the conclusion is given in section 5.

2. RELATED WORK

We should introduce AQM, service discovery protocols, and the relevant work and algorithms in order to understand the subsequent sections. In fact queues are a solution for the asynchronous flow of packets or operations and the used AQM influences the protocols which depend on sending messages like service discovery protocols. From this point of view, it is a significant to introduce service discovery protocols and AQM to understand the proposed algorithm properly.

2.1 Service Discovery Protocols

Service discovery protocols enable devices to discover all services in a network and some of them allow devices that provide services to announce their services. Each service discovery protocol must have two components: a client which is the component that has a set of requirements that form the services it needs, and a device which is the component that offers its service(s) and is requested by client. Accordingly, any node in a network may be a client, a device, or a client and device at same time. Service discovery protocols can be classified into two types: Registry-based such as Jini [6][7] and Peer-to-Peer like UPnP [11]. Registry-based can be further classified into centralized registry like Jini and distributed registry like Service Location Protocol (SLP) [6][8]. The Registry-based and Peer-to-Peer approaches both have advantages and drawbacks. For example: Registry-based is well organized and managed, but the registry node could cause a bottle neck problem for the entire network since if this node is damaged for any reason the clients are not able to access the required services. While in the Peer-to-Peer type all services send messages regularly even if there isn't a target client and this causes an unnecessary consumption for the networks' resources. Some protocols consider the announcement as an essential principle in service discovery issue such as UPnP whereas others protocols do not use the announcement approach such as Bluetooth [6][9]. A selection technique should be used to select the most appropriate service when the discovery phase results in two or more identical services. There are two selection modes: manual and automatic modes. In manual mode, service selection is the responsibility of the user entirely. This mode has drawbacks: users may not know enough about the services to distinguish among them and too much user involvement causes inconvenience. This mode is applied in all the investigated service discovery protocols. In automatic mode, the service discovery protocol selects the service this simplifies client programs. On the other hand automatic selection may not be select the choice that user wants.

Each service discovery protocol has a specific features and philosophy which are different from other protocols. Here we will explain UPnP in more details as it is used in our simulation model.

2.1.1 UPnP

UPnP is proposed for use in small office and home environments and targets both device and service discovery. It has the capability of automatically assigning IP addresses to networked devices. The components considered in UPnP are control points (clients) which are optional and devices (offers service(s)). Service discovery in UPnP is based on the Simple Service Discovery Protocol (SSDP) [5]. SSDP was proposed to discover devices and services in a network easily, quickly, dynamically, and without any a priori knowledge. It uses HTTP over unicast and multicast UDP packets to define two functions: search the services of a network and announce the availability of services in a network. UPnP cannot scale well since it uses multicasting extensively (multicasting is used both for service advertisements and service requests) [6]. When a control point is connected to network, it starts requesting the required service(s) by sending multicast message over UDP transport protocol. The service(s) that match the required criteria respond by sending a unicast message to requested control point. Consequently, the control point gets information about the requested service. On the other hand, when the device is connected to network, it starts announcing its service(s) regularly by sending multicast message (over UDP). A control point must send a multicast request with method **M-SEARCH** when desires to search the network for devices. Control points that know the address of a specific device can also send unicast requests with method M-SEARCH, figure 1 explains the multicast and unicast request with method **M-SEARCH** [11].

M-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
MAN: "ssdp:discover"
MX: *seconds to delay response*
ST: *search target*
USER-AGENT: *OS/version* **UPnP/1.1** *product/version*

(a- Multicast)

M-SEARCH * HTTP/1.1
HOST: *hostname:portNumber*
MAN: "ssdp:discover"
ST: *search target*
USER-AGENT: *OS/version* **UPnP/1.1** *product/version*

(b- Unicast)

Figure 1. Multicast and unicast request with method M-SEARCH.

Search target (ST) value takes three options values: 1- (ssdp:all): Search for all devices and services.

2- (upnp:rootdevice): Search for root devices only.

3- (uuid:*device-UUID):*Search for a particular device using specific features.

Note that *device-UUID* specified by UPnP vendor. It is clear that the number of replied messages from the matching services is based on the ST value. When the ST variable takes (ssdp:all) value the replied messages will be more than when it takes (upnp:rootdevice) value.

As in the first case, the discovery process targets the whole network while in the second case; it targets specific devices (root devices) only.

2.2 Active Queue Management

Many AQM mechanisms have been proposed to manage the queues, such as DropTail, Random Early Drop (RED) and Random Exponential Marking (REM). In practice, most of the routers being deployed use the simplistic Drop Tail (FIFO) algorithm, which is simple to implement with minimal computation overhead. The router accepts and forwards all the packets that arrive as long as it has buffer space available for the incoming packets. If a packet arrives and the queue is currently full, the incoming packet will be dropped.

RED [3] is a queue management scheme that is intended to remedy the short comings of the drop tail algorithm. The router could notify connections of congestion either by dropping packets arriving at the router or by setting a bit in packet headers. RED is an early congestion notification because an arriving packet may be dropped before the queue is full. Depending on the queue length the dropping probability of RED is decided. RED maintains two thresholds: a low threshold in which all arriving packets are accepted and a high threshold when all arriving packets are rejected. RED is in the congested state in which packets are randomly dropped when buffer fullness is between the thresholds, [10].

REM [4] uses a specific definition of congestion measure and a different marking probability function that makes it different from RED. It attempts to match user rates to network capacity while clearing buffers, regardless of the number of users. In REM the dropping probability observed by a user depends in a simple and precise manner on the congestion measures, summed over all the routers in the path of the user.

All the proposed solutions are applied to the router side, but our suggested solution is applied to end nodes (any node at network that could be considered as a client or service nodes) and at application layer. The router has many functions to perform and by applying the protocol to the other nodes on the network the load in router side will be eased and this will have a positive impact on the whole network performance.

3. Motivation

Many service discovery protocols are based on sending messages to perform the discovery task, such as UPnP and SLP. Depending on this mechanism, discovering the services in the network could result in bursts that lead to dropping messages. In fact, in this case the dropped messages caused other issues such as incomplete discovery, which can increase the latency in performing the required tasks as the required services have not been discovered by the requested clients.

Hence bursts cannot be avoided but can be organized by the protocols which may cause them by balancing between the existing network resources and the size of bursts with the rate of bursts.

Each networks physical components have limited capabilities that must be considered in applying any protocol, to avoid any fault and get optimal results. When applying a service discovery protocol the physical network resources must be taken into consideration to avoid dropping messages.

The suggested algorithm resolves the problem of dropping messages during the burst mode caused by service reply messages to two consecutive requested clients or more. In addition, it presents a method that balances between the required queue space in routers and the services response rate to the requested clients taking into account the available bandwidth. It could

increase the required queue size in order to increase the sending rate and vice versa. Increasing the required queue size could put the network at risk as it may result in an increase in dropped messages especially if there is a cross traffic in the network. The required queue space must be fairly identified to avoid dropping messages and avoid the latency in services responses.

4. THE PROPOSED ALGORITHM

The aims of the proposed algorithm are: determine the required sending queue space and the required time for the routers to forward all messages contained in a burst before receiving the next burst of messages. This has been achieved by controlling the services response sending rate. The proposed algorithm includes the instruction and equations that explain the relation between the required queue sizes and the interval separating two consecutive bursts of messages, to avoid dropping messages. Regarding the Open Systems Interconnection model OSI, the algorithm was designed to be applied in application layer. It could be included in the protocols and applications codes, which may cause bursts to the network in their strategies.

The following rules must be applied to compute the sending queue size in each router or the space which is required to be available in the sending queue of each router at the sending time and calculate the best interval for each router. The algorithm was tested when the routers were connected in a decentralised configuration.

4.1 Decentralised Algorithm

This algorithm is for Decentralised network topology which its routers connected in Decentralised method.

4.1.1 Queue size Algorithm:

The Algorithm which is used to calculate the size of the sending queue for each router is illustrated in Figure 2. The values *m* and *n* represent the number of clients and services that connected to Routers R_i respectively, where i=1, 2 … No. of routers.

For i=1 to No. of routers
Start
If (R_i not connected to any node)
$SQsize_{Ri} = 2.$

else
$SQsize_{Ri} = Rim + Rin + 1.$
End

Figure 2. Pseudo code of queue size algorithm

4.1.2 Best interval Algorithm:

In the suggested network topology any chosen router will divide the network into two parts, left and right.

In figure 3 the best interval at Equation (1) guarantees that a specific router would forward all the received messages to their destination (clients) before receiving the next burst of messages. It can be developed and take into consideration the available queue size for the specified router, as it represents the shared space between all the clients (receivers) connected to that router so an

overlap between two or more consecutive burst of messages can be achieved in order to minimize the required interval. Note z is the number of candidate routers.

Where, $(T(x_k)) = \frac{Message\ Size\ of\ service_k}{Bandwidth\ that\ message\ would\ use}$

In Equation (1) the $(T(x_j))$ value is the biggest among $(Time(x_k))$, $k = 1,2, \ldots Large$. $(T(x_j))$ represents the time the message utilizes the link.

Let the candidate routers C_R_h where h=1, 2 ... z.
For h=1 to z
Start
Identify the two parts left and right of C_R_h.
Calculate the No. of services on left ($\sum_{k=1}^{Lrh}(Sl_k)$ –) and on right ($\sum_{k=1}^{Rrh}(SR_k)$).

If ($Lr_h > Rr_h$)
    Large = Lr_h
else
    Large = Rr_h
Calculate the best interval of CR_h :
$(BIC\_R_h) = \sum_{k=1}^{Large}(T(x_k)) + \sum_{k=1}^{gaps}(T(x_k)) - (T(x_j))$ . . . (1)
End for
BI=0
For each (BIC_R_h) values h=1, 2... z do the following:
Start
If (BI <BIC_R_h)
    BI=BIC_R_h
End

Figure 3. Pseudo code of determining best interval algorithm

$\sum_{k=1}^{gaps}(T(x_k))$**:** represents the number of message times during which a specific router doesn't receive any service messages from nearest router(s). Here the average message size and average bandwidth is used. When there is a service connected directly to the nearest router, it would need at least two message times to reach the evaluated router.

The following equation represents the Best Interval for any router:

$(BIC\_R_h) = \sum_{k=1}^{Large}(T(x_k)) + \sum_{k=1}^{gaps}(T(x_k)) - (T(x_j))$. . **(1)**. Where h=1, 2… z.

The identified queue size in the routers can be used to minimize the BI value. Overlapped space (OS) value represents this minimization. Figure 4 show how the Overlapped space (OS) is calculated:

Identify the router neighbor to chosen router and this will be *Rneighbour* .
Let RLarge = Rneighbour.

Let $SQsize_{RLarge}$ = sending queue size of RLarge
Calculate OS = $(SQsize_{RLarge} - f(S_{RLarge})) / f(C_{Ri})$. . . (2)

Figure 4 Pseudo code of OS calculation steps

Where $f(S_{RLarge}) = \sum_{k=1}^{n}(S_k)$ is number of all services that are connected to the RLarge and

$f(C_{Ri}) = \sum_{k=1}^{m}(C_k)$ is the number of all clients that are connected to the Ri. OS is measured by messages number and it represents the empty messages space in the sending queue of the chosen router.

Equation (1) could use the OS value and could be written as:

The best interval (BI) = $\sum_{k=1}^{Big} T(x_k) + \sum_{k=1}^{gaps} T(x_k) - T(x_j) - \sum_{k=1}^{OS} T(x_k)$. . . **(3)**

The question now, must each router in a network be evaluated in order to identify the best interval for entire network? And which interval would be used for the network? The answer is: Not all routers in a network must be evaluated instead some of them would be a candidate to be evaluated and the longest interval will be used at the end; because logically using the longest interval will avoid dropping messages at all other routers.

There are some conditions that help to identify which router will have the most impact in determining the best interval.

4.1.3 Choosing candidate router rules:

The following rules identify the router(s) that act as bottlenecks in the messages flow paths. These rules identify the router(s) that required more time to forward the received messages during bursts:

Identify the longest path between a service and a client. Then the router which is connected to this client must be selected.

Identifying the router that is connected to the largest number of clients and receives the largest number of services from one side of the network.

Identifying the router that is connected to one or more clients and located nearest the end of the network.

If the chosen router is connected to one client then the nearest router connected to client must also be chosen, in order to compare between two consecutive burst of messages that reach these routers consecutively.

In case that the two (or more) consecutive burst of messages were sent to the same client and this client is the lonely client connected to router, this means logically there are two (or more) receivers connected to that router and this should be taken into consideration in calculating the (OS) value.

One client may satisfy more than one of the previous conditions, in other words the client that has longest path with a service could be the same client that connected to a router which receives largest number of services and this wouldn't cause any problem.

All the candidate routers must be evaluated and the longest interval is the best interval for the network which would guarantee no losing messages.

4.2 Centralised Algorithm

This algorithm is proposed for the networks where its routers are connected in a centralised topology. All routers in this topology are connected to one central router.

4.2.1 Queue size Algorithm:

Figure 5 illustrates the algorithm which is used to calculate the size of the sending queue for each router. The values of *m* and *n* represent the number of clients and services that connected to *Ri* respectively, where i=1, 2 … No. of routers. *RLargeS* represents the router connected to the largest number of services and *RLargeSni* represents the number of services which connected to *RLargeS*.

Let RLargeS be the router that is connected to the largest number of services.
RLargeSni = number of connected services to RLargeS.
For i=1 to z
Start
If Ri is root router $SQsize_{\mathrm{Rroot}}$ *= n-(RLargeSni – 1).*
Else $SQsize_{\mathrm{Ri}} = f(C_{Ri}) + f(S_{Ri})$.
End for

Figure 5. Pseudo code of queue size algorithm

4.2.2 Best interval Algorithm:

It is clear that any sending message between any two routers must pass through the root router, in order to reach their destination. Figure 6 represents the best interval algorithm. Where *RLargC[i]*, i=1, 2...j is an array of the router(s) that connected to the largest number of clients and *RLargeC_S* is the router that connected to the largest clients number and lowest services number.

Where $f(C_{Ri}) = \sum_{k=1}^{\mathrm{m}}(C_k)$ is number of all clients that connected to the Ri.

and $f(S_{Ri}) = \sum_{k=1}^{\mathrm{m}}(S_k)$ is number of all services that connected to the and Ri.

Where, $(T(x_{\mathrm{k}})) = \frac{Message\ Size\ of\ service_{\mathrm{k}}}{Bandwidth\ that\ message\ would\ use}$

$\left(T(x_{\mathrm{j}})\right)$ value is the biggest among $(Time(x_{\mathrm{k}}))$, $k = 1,2,\ldots Large$. $\left(T(x_{\mathrm{j}})\right)$ represents the time the message utilizes the link.

$\sum_{k=1}^{\mathrm{gaps}}(T(x_{\mathrm{k}}))$**:** represents the number of message times during which a specific router doesn't receive any service messages from nearest router(s). Here the average message size and average bandwidth is used.

Identify RLargeC_S which is the router that connected to the largest number of clients and largest number of services.
The BI = $\sum_{k=1}^{\mathrm{n}}(T(x_{\mathrm{k}})) + \sum_{k=1}^{\mathrm{gaps}}(T(x_{\mathrm{k}})) - \left(T(x_{\mathrm{j}})\right) - \sum_{k=1}^{\mathrm{n}}(T(x_{\mathrm{k}}))$

Figure 6. Pseudo code of determining best interval

4.2.3 Choosing candidate router rule:

The following rule should be followed to choose the proper router that influences the length of the interval.

- Identify the router that connected to the largest number of clients and lowest number of services.

5. SIMULATION RESULTS

The applied simulated models clarify the influence of the AQM on performing of UPnP. It has been compared between the algorithm and normal cases over UPnP. The network design includes 4 routers (R0, R1, R2, and R3) connected in decentralized manner in first model and in centralized manner in second model where each router is connected to 3 services (S0, S1..., and S9) except R2 which connected to six clients (C0, C1... C5). Network parameters have been shown in table 1.

The applied scenario is:

1. (C0, C1... C5) send multicast messages to discover all the services in the network, then,

2. All services send reply messages to the requested clients. In algorithm each service separates any consecutive replying messages with a specific period of time. While, in normal case service replies to the discovery requested dependently.

There is a UDP background traffic (S0 with S8) and (S1 with S7), where S0, S1 are connected to R0 and S7& S8 are connected to R3. The rate of the backward traffic is 0.01 and the messages size is different.

Depending on the chosen candidate router rules, R2 is the candidate router. It is clear that the services which are connected to R6 have longest path to reach C0 and C1.

Table1. Network Parameters

Parameter name	Value
Bandwidth among routers (Main links)	512Kb
Bandwidth between routers and other nodes (Sub links)	256Kb
Delay in main and sub	0ms
Queue Type	Drop Tail
Routing Protocol	DSDV
Message Length of discovery (Multicast)	64 bytes
Message Length of discovery reply (Unicast)	128 bytes
Message Length of backward traffic	100, 200, 300 bytes
Simulation Time	100.0 seconds

5.1 DECENTRALIZED MODEL

In this model the router is connected in Decentralized mode. Figures (7, 8, 9, 10, 11 and 12) explain the impact of deploying the different AQM on the performing of suggested algorithm and normal case over UPnP.

Depending on the queue size algorithm, the queue size for all the routers except R2 is:

$SQsize_{R0}$ =4+1= 5 packets. While queue size for R2 = 2+1= 3 packets

To calculate the best interval the Overlapped space (OS) must be calculated

Overlapped space (OS): OS = (5-3)/4.

OS = 2 / 4, OS = 0.5, OS=0 messages spaces.

All replying messages are equal in size and bandwidth of the links is different so:

$T(x_k)$ could be two values.

$(T(x_k)) = \frac{256*8}{512*1024}$, $(T(x_k)) = 0.00390625 \approx 0.004$ seconds.

1- $(T(x_k)) = \frac{256*8}{1000*1024}$, $(T(x_k)) = 0.002$ seconds.

The best interval (BI) = $\sum_{k=1}^{12} 0.004 + \sum_{k=1}^{1} 0.004 - 0.004 - \sum_{k=1}^{0} 0.004$

$\Rightarrow$ (BI) = 0.048 seconds.

The flowing figures (7, 8, and 9) show the network utilization in main links when this interval is applied in NS2 simulator.

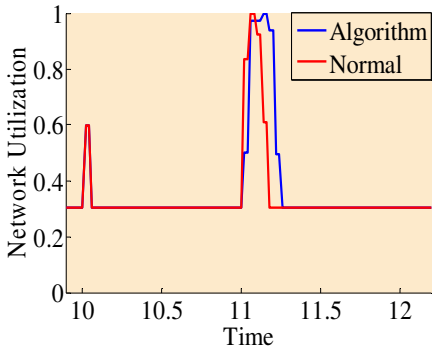

Figure 5. Droptail (Main links)

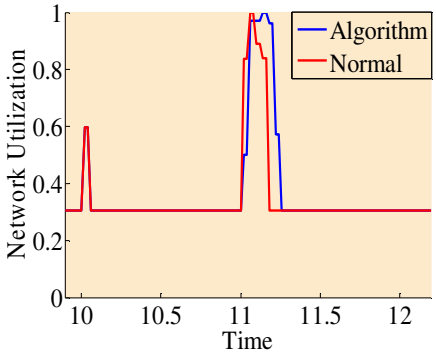

Figure 6. RED (Main links)

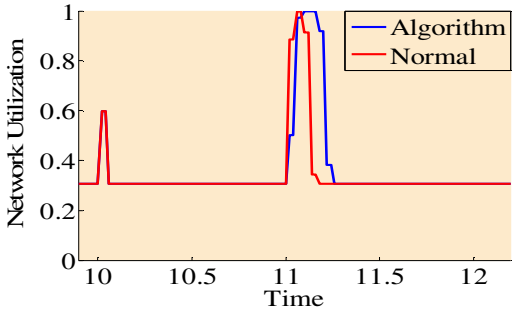

Figure 7 REM (Main links)

In main links the network utilization has been improved in Drop-tail, RED and REM where time specified for the burst of messages has been increased to be long enough to deliver all the messages of burst of messages before receiving the next burst of messages. Although, the Algorithm did its calculations but the network utilization in algorithm case reached the upper limit of the network which resulted in dropped messages. This is attributed to the backward traffic which the algorithm does not take into its' considerations.

In sub links the results show a clear difference between normal and proposed algorithm in Drop-tail, REM and RED, as explained in Figure 8, 9 and 10. Regardless the used AQM, in normal case the network utilization reach the upper limit with dropped messages, while in algorithm case the network utilization is below 85% without dropped messages.

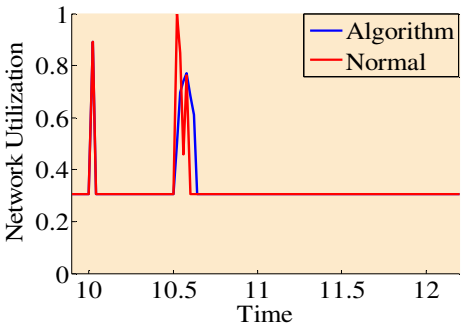

Figure 8. Droptail (Sub links)

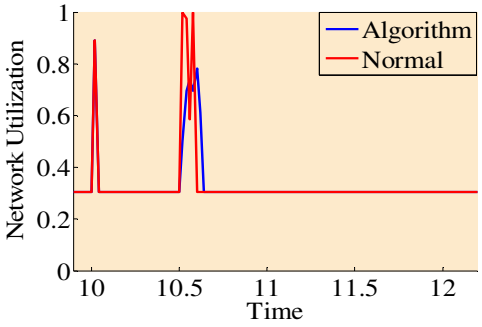

Figure 9. Red (Sub links)

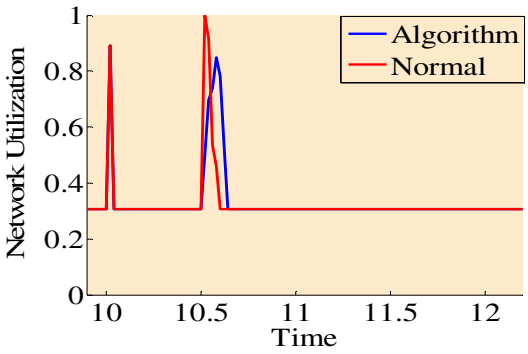

Figure 10. Rem (Sub links)

The previous figures show the proposed algorithm expands the discovery period of time by decreasing the services response sending rate to avoid dropping messages. Consequently, this guarantees the speed and efficiency in delivering all the messages to their destination without loss. In addition, these figures shows the impact of the backward traffic on network utilization and dropping messages and this must take into algorithm consideration to improve its performance.

Figures 11, 12, 13 explain the discovery rate in the tested AQMs. In general the discovery rates in algorithm are higher than normal case.

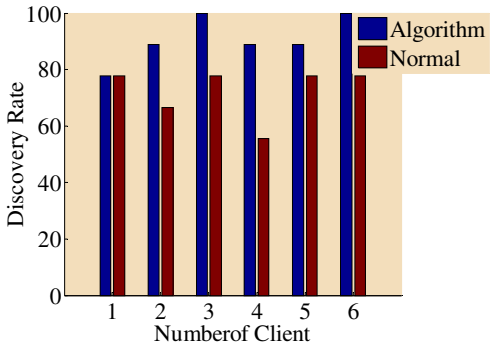

Figure 11. Discovery rate in Droptail

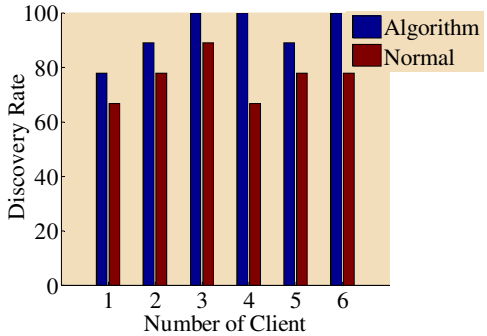

Figure 12. Discovery rate in Red

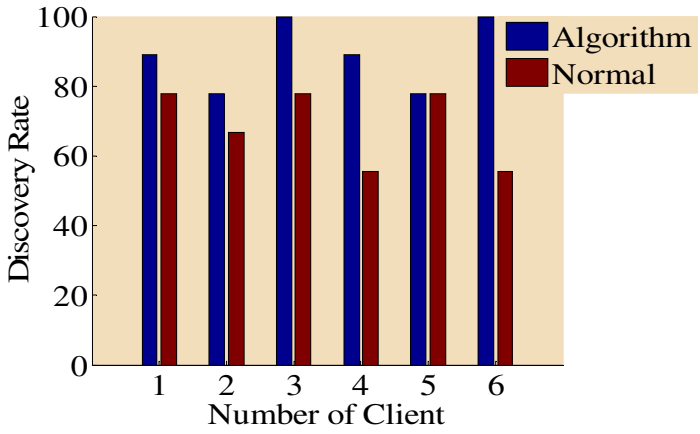

Figure 13. Discovery rate in Rem

In algorithm case, the discovery rate rang is between (77.77%) to (100%), while in normal case, the discovery rate range is between (55.55%) to (88.88%). Red has achieved the best discovery rate, as C3, C4 & C6 discovered the whole network, C2 & C5 discovered 88.88% of the services and C1 discovered (77.77%). Rem shows the worst discovery rate since C3 & C6 discovered (100%), C1 & C4 discovered 88.88% and C2 & C5 discovered (77.77%) of the services

In normal case the Red achieved the best discovery rate and Rem achieved the worst discovery rate too.

Figure 14 and 15 shows the Dropping rate in examined AQM when the algorithm is deployed and in normal case.

Depending on the chosen candidate router rules, R2 is the candidate router. It is clear that the services which are connected to R6 have longest path to reach C0 and C1.

Where the dropping rate is calculated based on the number of all the sent messages during the period of sending the services' reply messages including the cross traffic messages.

In the algorithm case, the worst number of dropped messages is in Droptail and REM protocols. While without using the algorithm the worst case is in REM protocol protocols.

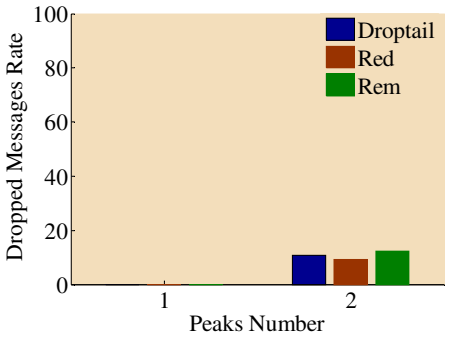

Figure 14. Dropping rate in AQM at Algorithm case

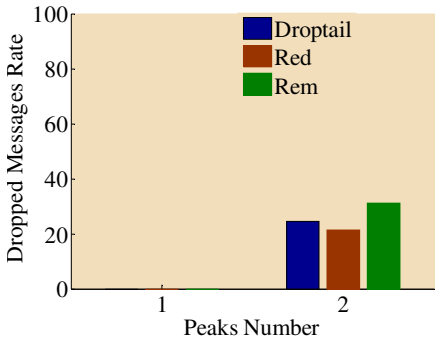

Figure 15. Dropping rate in AQM at Normal case

In general the algorithm shows significant improvement in the results, in terms of network Utilization, discovery rate and dropping rate when it deployed with RED protocol.

5.2 CENTRALIZED MODEL

In this model additional router (R4) is add to the topology of the suggested network to acts as the central node that all the other routers should connect to it. Figures (16, 17, 18, 19, 20 and 21) explain the impact of increasing the consumption of network resources such as bandwidth on the performing of suggested algorithm and normal case over UPnP.

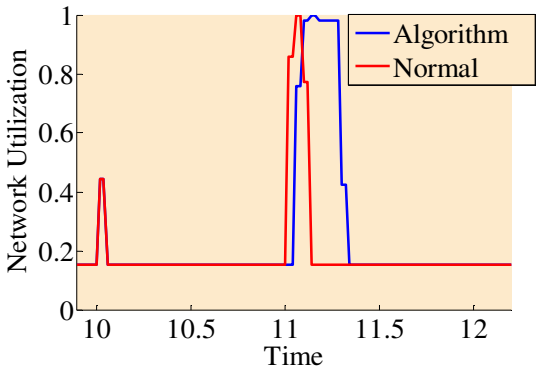

Figure 16 Drop tail (Main links)

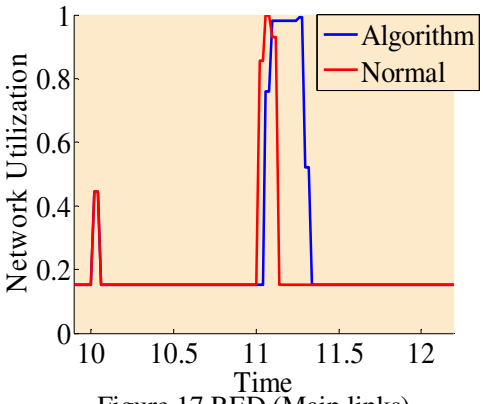

Figure 17 RED (Main links)

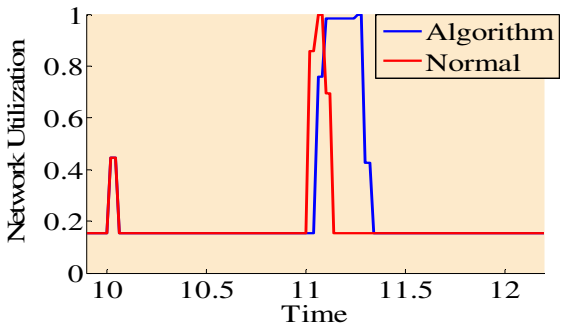

Figure 18 REM (Main links)

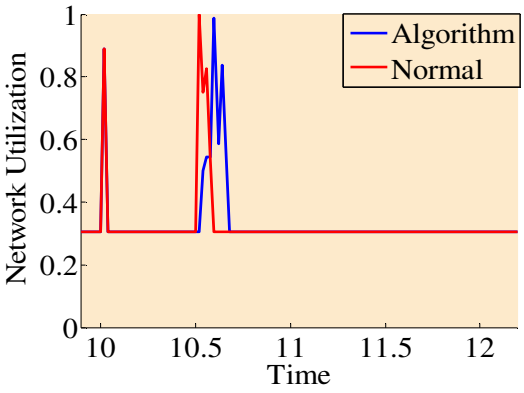

Figure 19 Drop tail (Sub links)

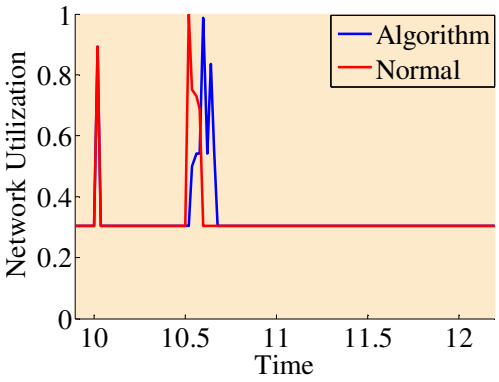

Figure 20 Red (Sub links)

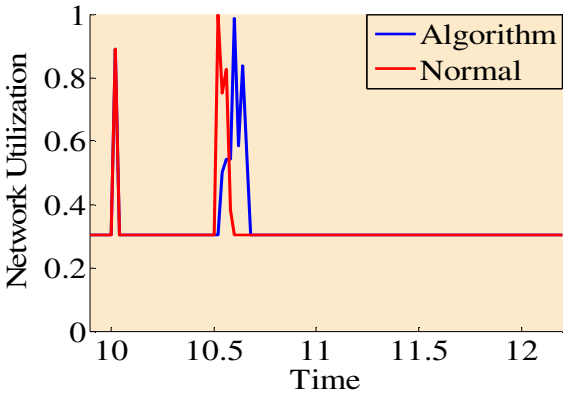

Figure 21 REM (Sub links)

In general, the network utilization at main and sub links has been improved when the algorithm is deployed as it decrease the sending rate of replied services messages and this allow more time to deliver these messages and avoid dropping them. As mentioned previously the algorithm did not take the cross traffic in its calculations resulting in dropped messages.

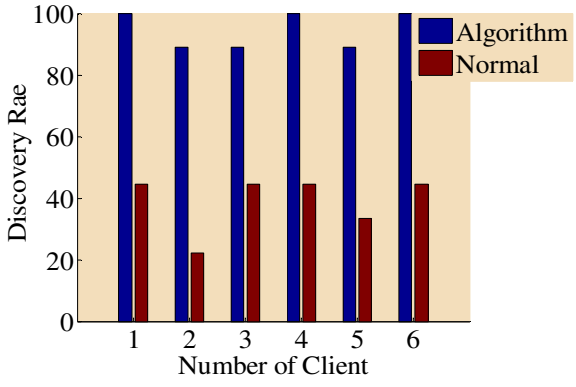

Figure 22. Discovery rate in Droptail

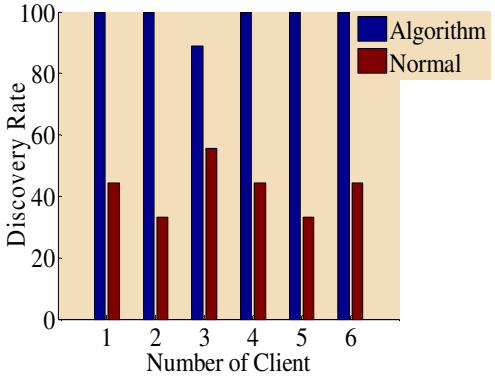

Figure 23 Discovery rate in Red

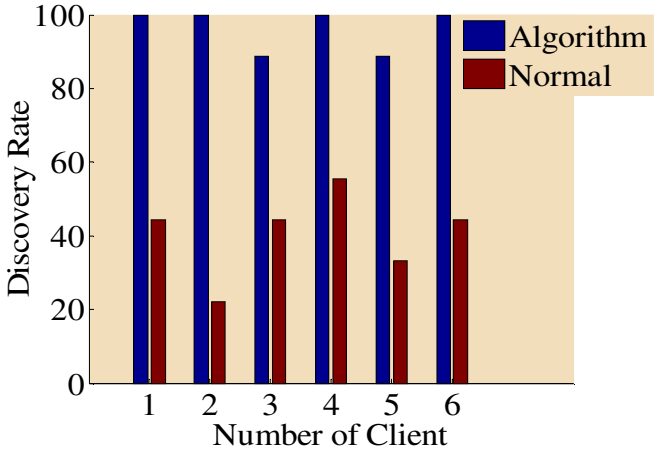

Figure 24. Discovery rate in Rem

Figures 22, 23 and 24 represent the discovery rate in examined AQM in two cases algorithm and normal. The discovery rate range between (88.88%) to (100%) in algorithm case, while it range between (22.22%) to (55.55%) , in normal case. Red has achieved the best discovery rate, as C1, C2, C4, C5 & C6 discovered the whole network, C3 discovered 88.88% of the services. Drop tail shows the worst discovery rate since C1, C4 & C6 discovered (100%), C2, C3 & C5 discovered 88.88% of the services.

In normal case the Red achieved the best discovery rate and Drop tail achieved the worst discovery rate too.

The dropping rate in examined AQM is shown in figure 25 and 26 in two cases algorithm and normal.

Where the dropping rate is calculated based on the number of all the sent messages during the period of sending the services' reply messages including the cross traffic messages.

In the algorithm and normal cases, the worst number of dropped messages is in Drop tail and REM protocols and the Red achieved the least number of dropped messages.

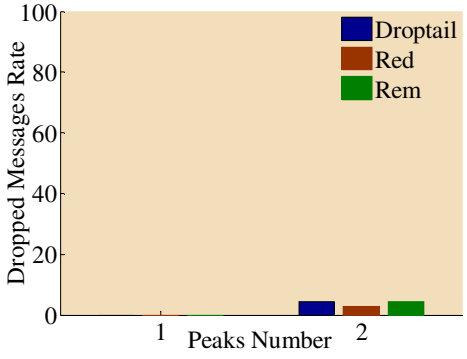

Figure 25. Dropping rate in AQM at Algorithm case

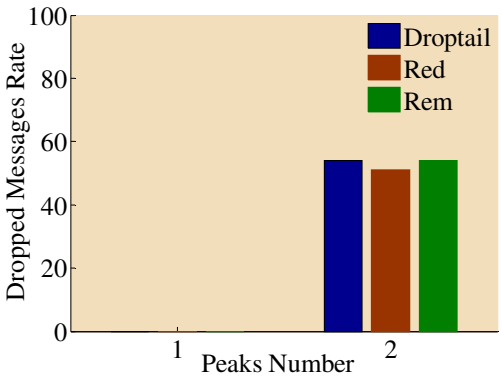

Figure 26. Dropping rate in AQM at Normal case

Generally, the achieved results by the suggested algorithm show improvement in centralized networks more than in decentralized networks.

6. CONCLUSION

Dropping messages during service discovery protocols results in incomplete discovery process, which causes delay and inefficiency in connected nodes(s) performance. The proposed algorithm addressed this issue by separating consecutive burst of messages with proper interval of time, as this would give network the required time to deliver all the burst of messages before receiving the next burst of messages. The proposed algorithm introduces a new method to determine the relation between the required available spaces in router queue with the needed interval between two consecutive burst of messages. It suggests specific approaches to calculate the required sending queue size, best interval and choosing candidate routers for the decentralized and centralized network. These approaches can be modified easily in order to reduce the best interval by increasing queue size and vies versa. This algorithm takes into account the network configuration beside the bandwidth and the message size, which impacts the sending and receiving rates.

The suggested algorithm applied on three different AQM protocols which are Drop-tail, RED and REM. The results from previous experiments show improving in the network utilization, discovery rate and dropping rate when the suggested algorithm is deployed with the three AQM protocols. Further, they show that the suggested algorithm is influenced by the proportion of consumed network resources and this reveal the need for providing additional/different parameters in the suggested algorithm to cope with the changing in the available network resources.

REFERENCES

[1] Al-Mejibli, I and Colley, M, "Evaluating Transmission Time of Service Discovery Protocols by using NS2 Simulator", Wireless Advanced (WiAD), 2010 6th Conference on London.

[2] T. Bhaskar Reddy and Ali Ahammed, "Performance Comparison of Active Queue Management Techniques", Journal of Computer Science 4 (12): 1020-1023, 2008

[3] Floyd, S., and Jacobson, V. "Random Early Detection gateways for Congestion Avoidance" V.1 N.4, August 1993, pp. 397-413.

[4] S. Athuraliya, Victor H. LiSteven H. Low, Q. Yin, "REM: Active Queue Management", January 15, 2001

[5] Ting Cai, Paul Leach, Yaron Y. Goland, and Shivaun Albright, "Simple Service Discovery Protocol/1.0", Internet Engineering Task Force "http://tools.ietf.org/html/draft-cai-ssdp-v1-01", April 8, 1999.

[6] Christopher N. Ververidis and George C. Polyzos, "Service Discovery for Mobile Ad Hoc Networks: A Survey of Issues and Techniques", "http://mm.aueb.gr/publications/2008-SD-SURVEY-COMST.pdf.", 14 November 2006,

[7] Michel Barbeau, Evangelos Kranakis and Honghui Luo, "Strategies for Service Discovery over Ad Hoc Networks", "http://www.engineeringletters.com/issues_v13/issue_1/EL_13_1_2.p df", 4 May 2006.

[8] "Novell Documentation: Novell eDirectory 8.7 - How SLP Works", "http://www.novell.com/documentation/edir87/?page=/documentation /edir87/edir87/data/a60jiyy.html".

[9] Eugene A. Gryazin, "Service Discovery in Bluetooth",Helsinki University of Technology.

[10] S. Salah and M. Fleury, "WiMAX Uplink Video Streaming Behavior", Proceedings of MoMM 2009.

[11] UPnP FORUM, "UPnP Device Architecture 1.1", UPnP FORUM members, 15 October 2008.