International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-2 Issue-5, November 2013

Enhancement of Back off Algorithm in CSMA/CA

Vijaya Sagvekar, Vidya Sagvekar

Abstract: ***Mobile ad hoc networks (MANETs) are useful in environment where fixed network infrastructure is unavailable. To function normally, MANETs demand an efficient and distributed medium access control (MAC) protocol. However, characteristics of MANETs such as radio link vulnerability, mobility, limited power pose great challenges on MAC design. Through this report, firstly identifying the challenges that are facing MAC in MANETs. Then presenting discussion on the problems with IEEE 802.11, especially those relevant in a singlehop and multihop networks and various techniques that have been proposed to enhance the channel utilization of wireless networks.***

Index Terms: medium access control (MAC),mobile ad hoc networks (MANETs), quality of service (QoS),fairness, energy-efficiency.

I. INTRODUCTION

Mobile Ad-Hoc Networks (MANETs) have achieved a large amount of growth in recent years. The IEEE 802.11 access protocol which was originally designed for wireless local area networks has been invoked repeatedly in the context of MANETs to provide better performances in these networks. In general in such scenarios, wireless nodes have a shared channel and every node should compete for the channel before it can send its own packet. The IEEE 802.11 provides detailed Medium Access Control (MAC) and physical layer (PHY) specification for wireless networks. IEEE 802.11 MAC contains two coordination functions, namely, Point Coordination Function (PCF) and Distributed Coordination Function(DCF) which support the infrastructure and Ad-Hoc configuration. PCF depends on a central coordinator to allocate channel resource and provide services without any competition. While DCF is a mandatory and contention-based protocol. In MANETs, due to lack of access points, contention-based distributed channel access protocols are more efficient. Distributed contention-based shared algorithms in ad hoc network have been mainly focused on achieving fairness and increasing spatial channel reuse. Using timestamp-based mechanism for assigning packet to transmit. However, most of these efforts focused on achieving fairness. Not considering the problem of bottlenecks. However, the bottleneck nodes will be the key point to degrade the throughput performance in ad hoc network. Through this paper [2], introducing the new algorithm which coordinates between immediate neighbor links of the same multi-hop flow and assigning contention window (CW) according to that neighbor links status.

Revised Manuscript Received on 30 November 2013.
* Correspondence Author
Vijaya R. Sagvekar received B.E. degree in Computer Engineering from the University of Mumbai, and M.Tech in Information Technology from the NMIMS University, Mumbai, and Maharashtra, India.
Vidya R. Sagvekar received B.E. degree in Electronics Engineering from the Pune University, in 2004, and M.E. in Electronics and Telecommunication Engineering from the Mumbai University, Mumbai, and Maharashtra, India.

Discussing two new backoff procedures that will reduce waiting time before transmitting any packet. These two algorithms will increase the throughput performance in ad hoc network.

[1]If the number of contending stations is known, the backoff mechanism can be tuned to attain the optimal performance of CSMA/CA. However, the estimation of the number of contending stations is not a trivial task. Under the assumption of ideal channel conditions and saturated stations, advanced filtering techniques can be used to accurately estimate the number of contenders. The estimation of the numbers of contenders is even more challenging when the saturation assumption is released. Unsaturated flows join the contention to transmit only one packet and hence, in most of the occasions, they manifest selves only at the instant they stop competing for the channel. It is proposed to modify the way that contention windows grow and shrink. Another approach is not to choose the backoff randomly, if the stations are aware of the backoff values of the other contenders, collisions can be effectively avoided.

The aforementioned approaches exhibit one or more of the following weaknesses:

a) They rely on the saturation assumption or on the ideal channel assumption

b) They require a modification of the packet headers

c) They cannot fairly coexist with legacy DCF.

By using a deterministic (and equal for all stations) backoff after successful transmissions, the collisions in the WLAN are significantly reduced, and even disappear. The reason is that collisions cannot occur among those stations that successfully transmitted and chose the same deterministic backoff value. Using the name CSMA with Enhanced Collision Avoidance (CSMA/ECA) to refer to the new protocol that uses a deterministic backoff after successes.

II.MEDIUM ACCESS CONTROL CHALLENGES AND DESIGN ISSUES

While MANETs(Mobile Ad-hoc Networks)[6], exhibit unique advantages compared to one-hop wireless networks such as cellular networks and wireless local area networks (WLANs), they do impose several challenges and design issues on MAC protocol design. The first and most serious challenge is that centralized controlling usually is not available in MANET due to the lack of infrastructure support. Without perfect coordination, collisions could take place when several nodes simultaneously access the shared medium. They may also result from transmissions that are multiple-hop away. Second, due to hardware constraints, a node cannot immediately detect collisions during ts transmission, which leads to channel inefficiency.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Third, as every node in the network is mobile, the network topology may change from time to time. Accordingly, each node may experience different degree of channel contention and collision. At the same time, the attendant route changes also affect the interaction between the MAC layer and higher layers. Finally, several important issues like energy efficiency, fairness, or quality of service (QoS) provision need to be carefully considered when designing MAC protocols for MANETs. Here summarizing some challenges in MAC Design:

1. Collisions:

Collisions come from two aspects in MANETs. They may occur due to simultaneous transmissions by two or more nodes in a certain range where their signals collide and interfere with each other. Obviously, the more the active nodes in the range of a transmitter receiver pair, the more severe the collisions observed. On the other hand, collisions can result from hidden terminals. A hidden terminal is the one that can neither sense the transmission of a transmitter nor correctly receive the reservation packet from its corresponding receiver. In the IEEE 802.11 MAC protocol, the reservation packet is a clear-to-send (CTS) packet, which advertises the reservation of the channel. A hidden terminal node can interfere with an ongoing transmission by transmitting at the same time.

2. Fairness:

Unfairness could result from different opportunities of channel access. In MANETs, there are two major sources for unequal channel access opportunities: the backoff mechanism and location. While the backoff mechanism is widely used in MAC protocols for MANETs to reduce collisions and achieve high channel efficiency, it always favors the node that just successfully seized the channel. As a result, different nodes may use different backoff window, leading to different transmission probabilities and consequently short-term unfairness as well as long term unfairness. Meanwhile, since nodes' location and traffic might not be uniformly distributed in MANETs, a node's location also influence its channel access opportunity. Nodes with less channel contention from their neighboring nodes can seize the channel more likely than others. Note that to achieve fairness among all nodes, the network's aggregate throughput, namely, efficiency, often has to be sacrificed.

3. Energy Efficiency:

In wireless networks, energy efficiency is always a critical issue due to a limited battery life. First, MAC protocols should reduce the number of collided packets as many as possible, and hence reduce the power consumption wasted in collisions. Second, only just enough power should be used to achieve a certain data rate for each transmission while maintaining good coordination among all the nodes. In addition, less transmission power can also reduce interference to other ongoing transmissions and improve the spatial reuse.

4. Quality of Service (QoS):

With the proliferation of Internet multimedia services, such as voice over IP and streaming video, mobile devices in MAENTs are expected to support these multimedia services with QoS guarantee. Since multimedia services typically have strict end-to-end delay and delay variation requirements, QoS provisioning will not be easy given that MANETs are characterized by their distributed and bandwidth-limited channel access, where medium contentions and collisions are common.

A.CSMA/CA

The Medium Access Control (MAC) is the mechanism that arbitrates the sharing of the channel among competing stations. In IEEE 802.11 networks, the MAC layer employs a combination of Carrier Sense Multiple Access and Collision Avoidance (CSMA/CA). The resultant protocol is called Distributed Coordination Function (DCF), and its behavior significantly impacts the overall performance of the network.

The stations running CSMA/CA sense the channel for ongoing transmissions before sending a packet. A station is allowed to transmit only if it senses the channel idle. It may happen that two or more stations begin a transmission (almost) simultaneously and a collision occurs. In order to reduce the chances of collision, the channel time is divided in slots and the transmissions are deferred a random number of slots.

The backoff values (B) are chosen from a contention window:

$$B \sim u\,[0, \min(CWmin \cdot 2^{\alpha}, CWmax) - 1] \qquad (1)$$

where u represents the uniform distribution. CWmin and CWmax are the minimum and maximum contention windows, respectively. The number of transmission attempts for the current packet is denoted as a (It equals 0 for the first transmission attempt).

The contention window uses a minimum value CWmin for the first transmission attempt and doubles after each failed transmission attempt, up to a maximum value of CWmax. This binary exponential growth reduces the number of transmission attempts in a congested scenario.

It is common to use a simple two-way handshake mechanism in which the data is transmitted in one packet and acknowledged by the receiver in a second packet. This modality is called Basic Access (BA).

There is an optional four-way-handshake floor-reservation mechanism to minimize the channel time waste due to collisions and prevent the hidden terminal impairment [11]. Request-To-Send and Clear-To-Send (RTS/CTS) packets are used before the actual data transmission in order to reserve the channel. When RTS/CTS is in use, collisions can only occur among control (short) packets, thus the amount of channel time wasted in collisions is reduced. However, the additional control packets penalize the overall efficiency of the network.

DCF is basically a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

Mechanism DCF defines two channel access methods: basic mode and RTS/CTS access mode. According to DCF, a node wishing to transmit packets senses the channel, if the channel is busy then it defers.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

If the channel is free for a specific time then the node is allowed to send. Before sending a data packet, a node should send a RTS (Request-To-Send) message to inform the receiver and other neighboring nodes that it wants to start a new communication. If the destination node replies it with a CTS (Clear-To-Send) message, it can start data transmission.

If the number of contending stations is known, the backoff mechanism can be tuned to attain the optimal performance of CSMA/CA. However, the estimation of the number of contending stations is not a trivial task. Under the assumption of ideal channel conditions and saturated stations, advanced filtering techniques can be used to accurately estimate the number of contenders.

The estimation of the numbers of contenders is even more challenging when the saturation assumption is released. Unsaturated flows join the contention to transmit only one packet and hence, in most of the occasions, they manifest themselves only at the instant they stop competing for the channel.

By using a deterministic (and equal for all stations) backoff after successful transmissions, the collisions in the WLAN are significantly reduced, and even disappear. The reason is that collisions cannot occur among those stations that successfully transmitted and chose the same deterministic backoff value. CSMA with Enhanced Collision Avoidance (CSMA/ECA) to refer to the new protocol that uses a deterministic backoff after successes.

B.CSMA/ECA

CSMA with Enhanced Collision Avoidance (CSMA/ECA), behaves exactly the same as the CSMA/CA protocol with the exception that a deterministic backoff is chosen after successful transmissions. To guarantee a fair coexistence with legacy CSMA/CA stations, the value of the deterministic backoff has to be:

$$V = \lceil E\left[\mathcal{U}[0, CW_{min} - 1]\right]\rceil = \lceil (CW_{min} - 1)/2 \rceil \quad (2)$$

Where [.] is the ceiling operator and E[·] is the expectation operator. The deterministic backoff after successes is a key parameter of the system, since it is also the maximum number of stations that can be accommodated in the collision-free mode of operation of CSMA/ECA. This parameter can also be adjusted to attain prioritization properties or to accommodate more contenders. For the first packet transmission and for transmission attempts following an unsuccessful transmission, the random backoff B as defined in [1] is used.

The backoff behavior of CSMA/ECA can be summarized as:

$$\text{backoff} = \begin{cases} V & \text{after a successful transmission;} \\ B & \text{otherwise.} \end{cases}$$

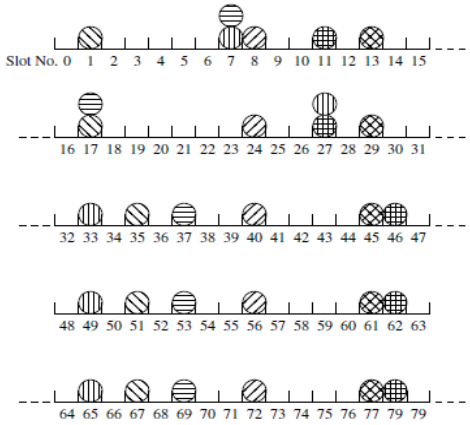

Fig1. A ball represents a transmission attempt in a given slot. Different filling patterns have been used to differentiate the transmissions of different stations. In CSMA/ECA the stations that successfully transmit use a deterministic backoff value.

Fig. 1 shows an example in which six CSMA/ECA saturated stations contend for the channel. The channel time is divided in numbered slots and the transmissions are represented as balls on that slot. The balls are filled with different patterns, each pattern corresponding to a different station. If there is only one ball in a slot, it is a successful slot. In contrast, if there are two balls in the same slot, a collision has occurred. The two stations involved in the collision will randomly choose a backoff value. It can be observed that there is a collision in slot number 7. The two stations choose backoff values 10 and 20, leading to two new collisions in slots 17 and 27, respectively. A station that successfully transmits, backoffs for V = 16 slots. As an example, the station that successfully transmits in slot number 13 in Fig. 1, also transmits in slots 29, 45, 61 and 77. It is useful to define the columns in the figure. A column is a set of slots whose numbers are equal modulo V (e.g. slots 0, 16, 32, 48 and 64 belong to the same column). Then, it can be observed that those stations that successfully transmit use the same column in their next transmission attempt. After all stations have successfully transmitted, in the slots numbered from 32 to 47, the behaviour of the system becomes deterministic and collisions disappear.

C .Comparison

[5]By choosing a deterministic backoff after a successful transmission and a random backoff otherwise, the system converges to a collision-free operation when the number of active stations is not greater than the value of the deterministic backoff. In the case of a successful transmission, the deterministic behavior stabilizes the system. Conversely, if there is a collision, the randomness of the backoff provides a change that would (desirably) avoid more collisions. CSMA/ECA exploits the information gathered from previous transmission attempts to further reduce the collisions, by performing a random search to find free slots until collisions disappear. Then the station keeps using a deterministic backoff after each successful transmission, until a collision occurs and the station moves back to the random behaviour.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Note that this collision have to necessarily be caused by a station using random backoffs, since collisions among stations that behave deterministically are not possible.

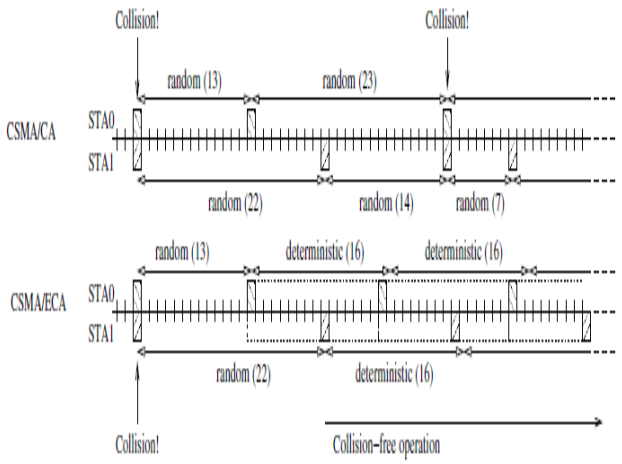

Fig. 2.[5] CSMA/CA is compared to CSMA/ECA

in an example in which two saturated stations contend for the channel. When CSMA/ECA is used, after both stations have successfully transmitted, the behaviour of the stations is deterministic and no more collisions occur.

Consider the case depicted in Fig. 2, where (STA 0 and STA 1) contend for the channel. For simplicity, all the slots are represented with equal sizes. Also the transmission attempts are represented as shaded boxes. The backoff value chosen by each station is shown in brackets1. The upper channel time line corresponds to legacy CSMA/CA operation, while the lower one shows our proposed CSMA/ECA operation. In this example, the CSMA/CA stations collide, successfully transmit and eventually collide again. When CSMA/ECA is used, collisions disappear after all stations have successfully transmitted, because the backoff is selected deterministically. It is useful to imagine a virtual frame2 of V slots (represented with a dotted line in the figure) and observe that after a successful transmission, the stations transmit in fixed slot positions within the virtual frame.

Algorithm 1: CSMA/ECA
1 . $b \leftarrow u[0, CWmin - 1]$;
2 . while there is a packet to transmit do
3. a ← 0 ;
4. while a < A do
5. while b > 0 do
6. wait 1 slot ;
7. b ← b − 1 ;
8 . Attempt transmission ;
9. if success then
/* Deterministic backoff. */
10. b ← V ;
11. break ;
12. else
13. a ← a + 1 ;
/* fall to random backoff. */
b ← U[0, min(CW_{min} * 2^a,CW_{max}) − 1]

Algorithm 1 shows the protocol that is distributedly executed in each of the contending stations, where b is the backoff counter, CW_{min} and CW_{max} are the minimum and maximum contention windows respectively, a is the number of transmission attempts, A is the maximum number of transmission attempts, and V is the deterministic backoff value.

III. BACKOFF ALGORITHM IN MULTI-HOP SCENARIO

In [2] a single channel contention based medium access control (MAC) protocols, whenever more than one station or node tries to access the medium at the same instant of time, it leads to packet collisions. If the collided station tries to access the channel again, the packets will collide as the nodes are synchronized in time. So the nodes need to be displaced in time. To displace them temporally, a backoff algorithm is used (example binary exponential backoff (BEB)). For example, in BEB algorithm, whenever a node's transmission is involved in a collision with another node's transmission, both nodes will choose a random waiting time and wait for this amount of time before attempting again. If they are not successful in this attempt, they double their contention window and choose a random waiting time before transmitting again. This process will be repeated for certain number of attempts. If the nodes are not successful in their transmission after this limit, the packets will be dropped from their queue. IEEE 802.11 MAC protocol does not coordinate between nodes at different hops within a multi-hop flow. This noncoordinate result in two drawbacks. First, immediate neighbor links of the same multi-hop flow will become adversaries in channel contention. This will increase the collision rate and reduce throughput. The second drawback is that, this non coordination results in the bottleneck problem. The differences in channel access capability between bottleneck links and non-bottleneck links will result in packet dropping at bottleneck nodes due to limited buffer size. This results in wastes of channel resource and transmission power and overall system throughput degradation. Proposed solution to solve the above two problems, develop [2] algorithm based on the 802.11 MAC protocol. Upstream node will use information from downstream neighbor node to adjust its contention window. The main idea of this algorithm is a downstream node forwarding old packets will always have priority over an upstream node sending new packets. With this algorithm, contention will be reduced between nodes in the same flow.

To make those packets forwarded by a downstream node recognizable by its immediate upstream neighbor, we should piggyback some flow information in the MAC header of data packets. Each node j should keep a flow table to record all the flows passing through it. The table contains a list of flow records of which include flow i's source address s_i, destination address d_i next-hop address nh_i, last-sent sequence number i and last-heard sequence number $1h_i$. The next-hop address nh_i is the node address of flow i's downstream immediate neighbor. The last-sent sequence number is flow i's latest sequence number of the packet that has been sent or is being sent by node j. The last-heard sequence number is the latest sequence number of the packet sent by nh_i which has been overheard by node j.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

A. Back off Procedure:

After a DIFS idle time, if a station senses that the medium is idle for a slot, then the backoff time will decrease by a slot time (BT_{new} = BT_{Old} - aSlotTime). When its backoff time reaches zero, the station will transmit a packet. But in our proposed method, if there are [(CW_{min} +1)*2-1] consecutive idle slots, its backoff timer is decreased by a half in each idle slot. We will call this algorithm as Fast Decreasing Backoff (FDB) and new backoff can be calculated as follows $BT_{new} = BT_{old} - BT_{old}/2$

Another backoff procedure, that will use to calculate new backoff timer (B_{new}) after the medium is detected to be busy at any time during a backoff slot with a nonzero backoff timer,will random new backoff time by using the rest of the value of the backoff timer (B_{old}). Calling this algorithm as Double Random Backoff (DRB) and this algorithm can calculate as follows:

$B_{new} = rand(0,B_{Old})$

Through this paper[2], discussing a new contention-based MAC algorithm that can reduce the opportunity of occurring bottleneck nodes by using the number of packets that has been sent in next node and current node for assigning the contention window. Also introducing[2], two new backoff procedures that can reduce the waiting time before sending packet in each node.

IV. NS-2 A PROGRAMMING TOOL

NS (version 2) is an object-oriented, discrete event driven network simulator developed at UC Berkely written in C++ and OTcl. NS is primarily useful for simulating local and wide area networks. Although NS is fairly easy to use once you get to know the simulator, it is quite difficult for a first time user, because there are few user-friendly manuals. Even though there is a lot of documentation written by the developers which has in depth explanation of the simulator, it is written with the depth of a skilled NS user.

NS is an event driven network simulator developed at UC Berkeley that simulates variety of IP networks. It implements network protocols such as TCP and UPD, traffic source behavior such as FTP, Telnet, Web, CBR and VBR, router queue management mechanism such as Drop Tail, RED and CBQ, routing algorithms such as Dijkstra, and more. NS also implements multicasting and some of the MAC layer protocols for LAN simulations. Currently, NS (version 2) written in C++ and OTcl (Tcl script language with Object-oriented extensions developed at MIT) is available.

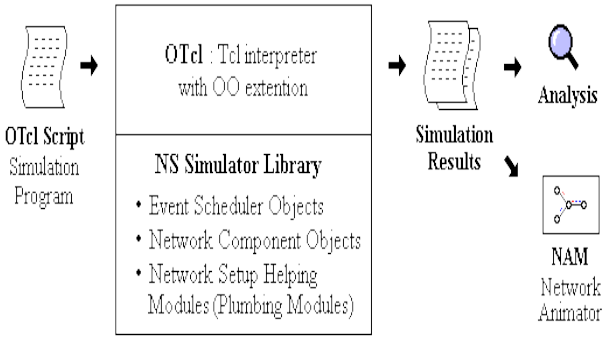

Fig3. Simplified User's View of NS

As shown in Figure 3, in a simplified user's view, NS is Object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component object libraries, and network setup (plumbing) module libraries (actually, plumbing modules are implemented as member functions of the base simulator object). In other words, to use NS, you program in OTcl script language. To setup and run a simulation network, a user should write an OTcl script that initiates an event scheduler, sets up the network topology using the network objects and the plumbing functions in the library, and tells traffic sources when to start and stop transmitting packets through the event scheduler. The term "plumbing" is used for a network setup, because setting up a network is plumbing possible data paths among network objects by setting the "neighbor" pointer of an object to the address of an appropriate object. When a user wants to make a new network object, he or she can easily make an object either by writing a new object or by making a compound object from the object library, and plumb the data path through the object. This may sound like complicated job, but the plumbing OTcl modules actually make the job very easy. The power of NS comes from this plumbing. Another major component of NS beside network objects is the event scheduler. An event in NS is a packet ID that is unique for a packet with scheduled time and the pointer to an object that handles the event. In NS, an event scheduler keeps track of simulation time and fires all the events in the event queue scheduled for the current time by invoking appropriate network components, which usually are the ones who issued the events, and let them do the appropriate action associated with packet pointed by the event. Network components communicate with one another passing packets, however this does not consume actual simulation time. All the network components that need to spend some simulation time handling a packet (i.e. need a delay) use the event scheduler by issuing an event for the packet and waiting for the event to be fired to itself before doing further action handling the packet. For example, a network switch component that simulates a switch with 20 microseconds of switching delay issues an event for a packet to be switched to the scheduler as an event 20 microsecond later. The scheduler after 20 microseconds dequeues the event and fires it to the switch component, which then passes the packet to an appropriate output link component. Another use of an event scheduler is timer. For example, TCP needs a timer to keep track of a packet transmission time out for retransmission (transmission of a packet with the same TCP packet number but different NS packet ID). Timers use event schedulers in a similar manner that delay does. The only difference is that timer measures a time value associated with a packet and does an appropriate action related to that packet after a certain time goes by, and does not simulate a delay. NS is written not only in OTcl but in C++ also. For efficiency reason, NS separates the data path implementation from control path implementations. In order to reduce packet and event processing time (not simulation time), the event scheduler and the basic network component objects in the data path are written and compiled using C++.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

These compiled objects are made available to the OTcl interpreter through an OTcl linkage that creates a matching OTcl object for each of the C++ objects and makes the control functions and the configurable variables specified by the C++ object act as member functions and member variables of the corresponding OTcl object. In this way, the controls of the C++ objects are given to OTcl. It is also possible to add member functions and variables to a C++ linked OTcl object. The objects in C++ that do not need to be controlled in a simulation or internally used by another object do not need to be linked to OTcl. Likewise, an object (not in the data path) can be entirely implemented in OTcl. Figure 2 shows an object hierarchy example in C++ and OTcl. One thing to note in the figure is that for C++ objects that have an OTcl linkage forming a hierarchy, there is a matching OTcl object hierarchy very similar to that of C++.

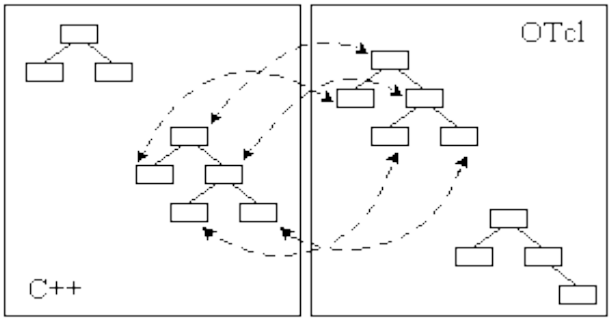

Fig.4. C++ and OTcl: The Duality

As shown in Figure 3, when a simulation is finished, NS produces one or more text-based output files that contain detailed simulation data, if specified to do so in the input Tcl (or more specifically, OTcl) script. The data can be used for simulation analysis (two simulation result analysis examples are presented in later sections) or as an input to a graphical simulation display tool called Network Animator (NAM) that is developed. NAM has a nice graphical user interface similar to that of a CD player (play, fast forward, rewind, pause and so on), and also has a display speed controller. Furthermore, it can graphically present information such as throughput and number of packet drops at each link, although the graphical information cannot be used for accurate simulation analysis.

V. CONCLUSION

The paper[1] explores the limits of the CSMA/CA random multiple access protocol, which is used in WLANs. It is shown that the performance of the protocols that randomly choose the slot at which transmission occurs is bounded by a fundamental trade-off. If the contenders aggressively transmit, the probability of collision is high.

Conversely, if the contenders use a low transmission probability (i.e. separate their transmission attempts by a large number of slots), the performance suffers because most of the slots remain empty. Although the transmission probability can be optimized, the resultant efficiency is still far from satisfactory. A conceptual change in the protocol is required to overcome the aforementioned fundamental bound.

Nevertheless, randomness is of paramount importance for resolving collisions. After a collision, it is desired that the implicated parts backoff for a different number of slots, in order to prevent that they collide in their next transmission attempt. Given the facts that random selection of the transmission slot limits the performance and that randomness is necessary to resolve collisions, a modification to CSMA/CA is proposed. It is suggested to use a deterministic backoff after successful collisions and a random backoff otherwise.

The immediate consequence is that, in saturation conditions, the stations that successfully transmitted in their last transmission attempt cannot collide in their next transmission attempt. Hence, the new protocol reduces the chances of collisions and thus it is named CSMA with Enhanced Collision Avoidance (CSMA/ECA). Moreover, if the number of contenders does not exceed the value of the deterministic backoff after successes, the systems converges to a collision free operation. After all the stations successfully consecutively transmit, collisions disappear. The suppression of collisions have a positive impact on the channel efficiency, which is the fraction of channel time devoted to successful transmissions. Actually, the performance of CSMA/ECA surpasses the efficiency upper bound associated to those protocols that randomly select the transmission slot.

In future, implementing protocol CSMA/ECA protocol under elastic, rigid and mixed flows.

Elastic flows are characterized by the fact that they have a clear tendency to consume all the bandwidth that is available in the network. They are typically associated to the use of the Transport Control Protocol (TCP) at the transport layer. At the MAC layer, they manifest as saturated stations. Web traffic, email, and peer-to-peer file interchange are good examples of elastic flows.

Rigid flows consume a fixed amount of bandwidth and are often encapsulated by the User Datagram Protocol (UDP) at the transport layer. During normal (uncongested) network operation, rigid flows do not saturate the station. On the contrary, the MAC queue remains empty for most of the time. A single packet is periodically received from the upper layer and, after the packet is serviced, the queue remains empty until a new packet arrives. Nevertheless, if the network is highly loaded and cannot transmit all the packets arriving from the upper layers, the MAC queues quickly build up and packet loss occurs due to queue overflow. If that is the case, we say that the network is congested. Voice over IP (VoIP) is an example of a service that uses rigid flows.

Through this paper[2], discussing a new contention-based MAC algorithm that can reduce the opportunity of occurring bottleneck nodes by using the number of packets that has been sent in next node and current node for assigning the contention window in multihop scenario.

REFERENCES

1. J.Barcelo, B. Bellalta, A.Sfairopoulou, C. Cano, M. Oliver",CSMA with Enhanced CollisionAvoidance: a Performance Assessment ",In Proceedings IEEE VTC Spring'09.
2. Somkiat Pornchaiwiwat and Watit Benjapolakul,"Coordinate Assigning Contention Window in Ad Hoc Network",TENCON 2006. IEEE Region 10 Conference.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

3. J.Barcelo, B. Bellalta, C. Cano, A.Sfairopoulou, M. Oliver, Carrier Sense Multiple Access with Enhanced Collision Avoidance: a Performance Analysis ",In Proceedings ACM IWCMC'09.
4. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 1999 Edition (Revised 2007).
5. Barcelo, J.Toledo, A.L. Cano, C.Oliver, M.," Fairness and Convergence of CSMA with Enhanced Collision Avoidance (ECA)",Communications (ICC), 2010 IEEE.
6. Hongqiang Zhai, Jianfeng Wang, Xiang Chen and Yuguang Fang," Medium access control in mobile ad hoc networks: challenges and solutions", Wireless Communications And Mobile Computing, 2006; 6:151–170 Published online in Wiley InterScience.
7. Ns-2manual: http://www.isi.edu/nsnam/ns

AUTHOR PROFILE

Vijaya r. Sagvekar received B.E. degree in Computer Engineering from the University of Mumbai, and M.Tech in Information Technology from the NMIMS University, Mumbai, Maharashtra, in 2012. Currently, she is an assistant Professor of Computer Engineering at University of Mumbai. Her teaching and research areas include Computer Networking, Computer Security, and Network Simulator.

Vidya r. Sagvekar received B.E. degree in Electronics Engineering from the Pune University, in 2004, and M.E. in Electronics and Telecommunication Engineering from the Mumbai University, Mumbai, Maharashtra, in 2013. Currently, she is an assistant Professor of Electronics Engineering at University of Mumbai. Her teaching and research areas includes Computer Networking, Network Simulator.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)