A NEW DISTRIBUTED CONTENTION CONTROL PROTOCOL FOR THE IEEE 802.11 MAC LAYER

Project of ATcrc Application Programms

Jiankun Hu, Konrad Eriksson

School of Computer Science & Information Technology, RMIT University, Melbourne 3001, Victoria, Australia.

Keywords: TCP protocol, MAC protocol, wireless networks

Abstract: The famous binary backoff algorithm in IEEE 802.11 MAC layer can forget the contention level between each successfully transmitted data frame and hence suffers significant performance degradation when the contention level is high. In IEEE 802.11 standard, a distributed contention control (DCC) algorithm is proposed to address this problem by observing how many of slots in the last backoff period were busy, i.e. slot utilization. The introduction of slot utilization can provide good estimation of the most recent congestion dynamics, i.e. transient fluctuations of the traffic. However, it is inaccurate to estimate the overall traffic pattern as one backoff period is too short to obtain accurate stationary statistics. In this paper, a new DCC algorithm is proposed that can combine transient and stationary characteristics, which provides better estimation of congestion level of the medium. Extensive simulation by using NS-2 simulator has shown that our scheme has better throughput and low collisions compared with original binary backoff MAC protocol and slot utilization algorithm (Nononi, Conti and Donatiello, 1998).

1 INTRODUCTION

Research on the wireless medium access protocol is an ongoing effort (Bononi, Conti and Donatiello, 1998; Weinmiller & Woesner, 1996; Weinmiller, Woesner & Wolisz, 1996; . Bianchi, Fratta & Oliveri,1996; Cali, Conti & Gregori,1998; Natkaniec & Pach, 2000). Most of the modifications to the backoff algorithm are done to make it adapt better to the current activity on the medium. Better adaptation makes the algorithm schedule transmissions in a way so that the medium is utilized to its maximum, realizing a throughput close to the theoretical limit. Because the original backoff algorithm in the IEEE 802.11 MAC layer is "forgetting" the contention level between each successfully transmitted data frame it suffers from performance degradation much below the theoretical limit when the contention level is high. Much of the degradation is due to the high number of collisions needed to increase the contention window (CW) size when the contention level is high in the medium.

One way to reduce the contention level, even if there are many stations competing for transmission, was introduced by H. Kim and J.C. Hou (Kim & Hou, 2003). They proposed to insert a delay before a STA attempts transmission of its pending frame. The length of the delay changes depending on how the medium is utilized, where the utilization level is estimated from the numbers of collisions registered between two consecutive successful data frame transmissions. Based on Goodman's work (Goodman, Greenberg, Madras & March, 1998), an analytical model for the medium access is established (Kim & Hou, 2003). The model defines two analytical components (the *fluid chunk* and the *MAC fluid*) that are used to characterize data transmission activities in the wireless medium.

A *fluid chunk* determines the number of currently active STAs with the number of collisions observed during the *fluid chunk*. The estimated number of STAs is then used to determine network utilization that is subsequently used for computing the scheduling delay. The normal access mechanism, defined in IEEE 802.11, is then used after passing the calculated delay. Simulation has demonstrated a substantial improvement in throughput by just adding this delay (Kim & Hou, 2003). This scheme still suffers the same problem, to a certain extent, by using number of collisions as this indicator lacks of temporal distribution information.

Hu J. and Eriksson K. (2004).
A NEW DISTRIBUTED CONTENTION CONTROL PROTOCOL FOR THE IEEE 802.11 MAC LAYER - Project of ATcrc Application Programms.
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 393-401
DOI: 10.5220/0001402903930401

L. Bononi *et al* (Bononi, Conti and Donatiello, 1998) introduced another way, called DCC, to improve the binary backoff scheme by adding an additional stage between the original scheduling part and the physical access. In this additional stage the scheduled transmission is evaluated again based on an estimated medium utilization value. The extended mechanism may defer the transmission at this stage with a probability based on how high the medium utilization is estimated. The deferring of transmission in the additional stage is fed back to the original transmission scheduler in the form of a virtual collision notification. The medium utilization is estimated every time the original mechanism backs off by counting how many of the slots during the backoff period were busy. The probability to transmit when the normal mechanism indicates transmission is then a probability function P_T that is based on the slot utilization (*Slot_Utilization*) and the number of attempts already performed (*Num_Att*). This function takes into account the number of attempts already performed to give STAs that have attempted and failed a higher priority to transmit. The more failed retransmissions the higher the probability becomes. This will generally reduce the maximum time for a transmission.

This mechanism does not modify the actual backoff algorithm proposed in IEEE 802.11(P802.11, 1997) but rather extends the original access mechanism. J. Weinmiller *et al* (Weinmiller & Woesner, 1996) examined the slot selection probability in the original backoff mechanism defined in IEEE 802.11(P802.11, 1997). They discovered that because every individual STA selects a slot within the contention window (CW) size with the same probability it turns out that early slots are more likely to be selected. The original mechanism says that STAs losing the competition have to pause their backoff countdown while the medium is busy. This results in that the slots close to transmission (early slots) are more crowded than slots in the end of the CW size, because they may contain both STAs that lost the last competition and new STAs that randomly selected at an early slot. Based on this observation, Weinmiller & Woesner, 1996) proposed a weighted slot selection to be used by each STA, where new STAs arriving in the competition have a higher probability to select a higher (later) slot. This spreads out the probability of a slot being busy equally over all slots in the CW. The problem is that every STA may have a different CW size and if they should communicate the size it would require additional bandwidth and complexity. Each STA is still resetting the CW size after a successfully transmitted data frame as the original algorithm does. This creates a high number of collisions if the contention level is high because of the consecutive collisions needed to increase the CW to the appropriate size.

F. Cali, M. Conti and E. Gregori model the dynamics of IEEE 802.11 MAC layer with Markovian chains (Cali, Conti & Gregori, 2000). They show that if a STA has exact knowledge of the medium status, it is possible to tune the backoff algorithm to achieve a protocol capacity very close to the theoretical maximum. Because exact knowledge of the medium status cannot be realized in real cases, they propose a way of adapting the contention window (CW) by observing the length of both the last idle period and the last transmission attempt. They use these two values to estimate the collision probability. The collision probability in combination with other calculated values ends up controlling the CW size. Both the length of the last idle period and the last transmission attempt are easy to calculate when modeling the behavior. However, the calculations introduced for estimating the number of STAs and medium load can be very heavy for devices with limited computing power. One way to solve the problem of dynamically estimating the unknown number of STAs is to keep it at a fixed value derived offline (Bruno, Conti & Gregori, 2002).

Up to now, almost all MAC protocols are based on the estimation of either transient network traffic dynamics or stationary statistics. In this paper, a new MAC protocol is proposed that can utilize both transient and stationary statistics, which provides better estimation of the network state, and hence improve network throughput. Extensive experiments have validated our scheme.

2 A NEW DISTRIBUTED CONTENTION CONTROL PROTOCOL

2.1 Collisions Average

To get a good estimation of the contention level of the medium we looked at the number of collisions that occur on the medium. Some collisions are unable to be avoided due to the randomness in selecting backoff slots at each STA, but too many collisions means that the contention level is higher than what the current contention window (*CW*) size Is optimal for. To get a good estimation of the congestion level, an average (*colAvg*) of the number collisions (*colCount*) that occurred during the last time interval (*colWindow*) was calculated. This is expressed as

$$colAvg = (colCount / colWindow) \qquad (1)$$

To count the number of collisions that occurred during this time interval, a history of collisions has to be maintained. The history of collisions is produced by having a buffer where new collision.

This history window, shown in Fig.1 can be implemented as a circular buffer. Whenever a collision occurs it is added to the buffer and the buffer is scanned for collisions that are outside the history window (*colWindow*). By using a circular buffer the old collisions are always at the end of the buffer making it easy to delete these by removing collisions at the end of the buffer until we encounter a collision event in the buffer with a timestamp that is inside the history time interval again. This saves scanning the complete history buffer and makes this history maintaining process a relatively low CPU intensive task.

By counting collisions and making the *CW* size depend on the number of collisions, the information about the contention level is also maintained between successfully transmitted frames avoiding letting the algorithm "try out" the contention from scratch again for each new packet. This is especially useful when the contention level is high and the normal algorithm would require several collisions and backoffs to occur before reaching the effective CW size.

2.2 Slot Utilization

This method was first introduced by L. Bononi (Bononi, Conti and Donatiello, 1998) and its basic idea is to sense the medium while backing off and therefore get a idea of the contention level before doing the next medium access.

According to the IEEE 802.11 specification(P802.11, 1997) a STA should always be able to sense the medium to detect if it is busy or idle. Further it also says that if the medium is detected busy during a slot while a STA is backing off the STA should not decrease its backoff counter at the end of that slot.

The slot utilization represents a measurement of the contention level of the medium during this backoff period. When the backoff period has ended, the normalized slotUtilization value is calculate as follows

SlotUtilization =NumBusySlots/InitBackoff (2).

InitBackoff is the randomly selected length of the backoff period and *NumBusySlots* is how many of the backed off slots were sensed busy. The *SlotUtilization* value is a measurement of how many of the slots that the STA spent on backing off . The value is normalized by dividing it with the total number of slots (*InitBackoff*) of the backoff period. This normalization changes the range to a decimal value between 0.0 and 1.0 where 0.0 means no slots were busy and 1.0 means that all slots were busy. This gives us a good indication of the contention level of the medium with little additional processing cost. The *SlotUtilization* value is a fast changing value because of the limited time period it samples over (one backoff period), which will gives us a value that represents a very current status of the system.

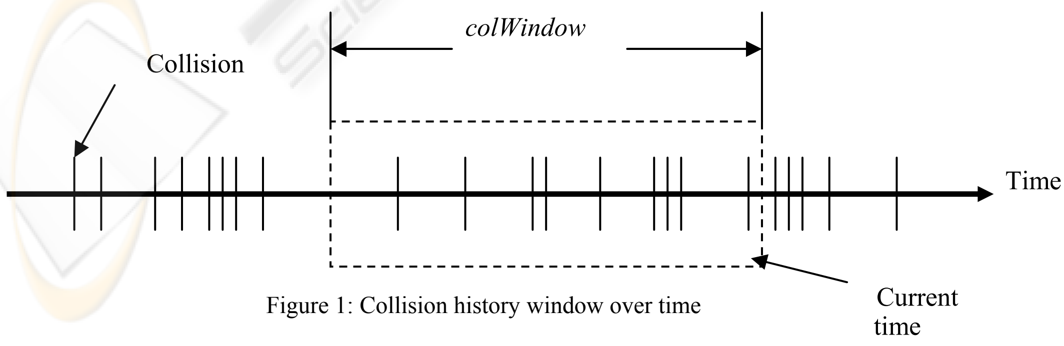

Figure 1: Collision history window over time

2.3 Combined Algorithm

To get the main benefits from both methods discussed, we propose the following formula to adapt the window

$$CW = colAvg + colAvg \times (SlotUtilization + K) \quad (3).$$

The collision average (*colAvg*, explained in section 2.1) is the slow changing average of the system representing a good overall picture of the contention level while the slot utilization (*SlotUtilization*, explained in section 2.2) gives the latest state of the medium making our algorithm adapt faster to sudden changes of the contention level.
This parameter *K* works as an offset for the second part of the equation, giving us the opportunity to decrease the *CW* size by setting *K* a negative value. The parameter *K* works as a threshold level where a *SlotUtilization* value under *–K* results in the *CW* size being decreased. The choice of *K* gives us the ability to adjust the CW size between 0.5 and 1.5 of the *colAvg*. How to select the optimal K is discussed in the subsequent sections. Our new algorithm can provide an accurate estimation of the overall congestion level while at the same time have the ability to quickly adapt to sudden changes in form of increased and decreased contention levels.

3 SIMULATION

3.1 Setup

The physical layer (PHY) was running as Direct Sequence Spread Spectrum (DSSS) and the attributes are given in Table 1. These attributes are taken directly from the recommendation in the IEEE 802.11 specification (P802.11, 1997). These values are used in all simulations unless otherwise stated.

Table 1: Attributes for IEEE 802.11 PHY in DSSS

Attribute	Value
CWMin*	31
CWMax	1023
SlotTime	20 us
CCATime	15 us
RxTxTurnaroundTime	5 us
SIFSTime	10us
PreambleLength	144 bits
PLCPHeaderLength	48 bits

*) The *CWMin* was set to 31 for all algorithms except the one proposed in this paper. The *CWMin* in our algorithm has a lower threshold, which the *CW* size can not go below even when the collision average and the slot utilization say otherwise. This is done to avoid a zero *CW* size.

The physical setup was that every node could hear each other (fully meshed network). The bandwidth of the physical medium is set to 2 Mbits. The layout of the data flows between the STAs is shown in Fig. 2. Each STA has two outgoing flows and two incoming flows in a ring fashion. This layout was chosen because it gives each STA the same amount of incoming and outgoing data. The ring layout could have been exchanged for a random destination model and give the same results. Once a STA has access to the medium it has the same probability of a successful transmission as any of the other STAs because they all share the medium.

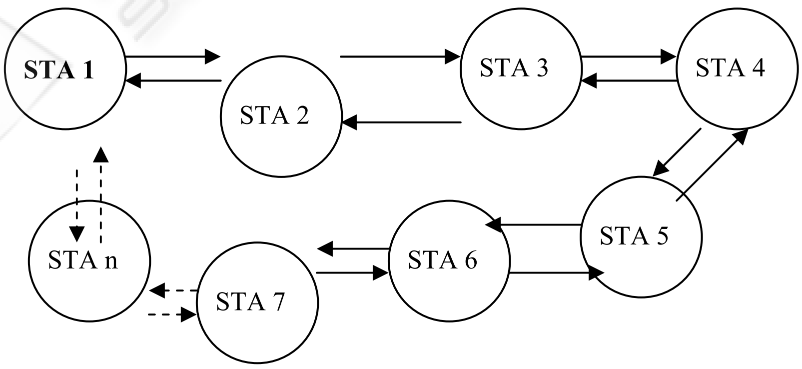

Figure 2: Data streams layout of simulations

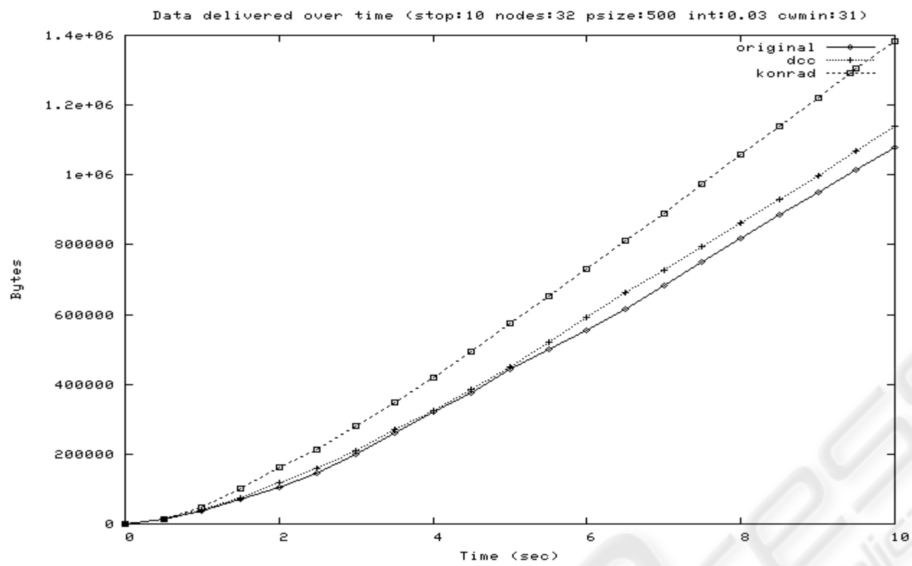

Figure 3: Data stream layout

3.2 Simulation Results

In Fig. 3 the accumulated data successfully transferred between all STAs are plotted against time. In this scenario 32 nodes were used and the simulation ran for 10 seconds for each algorithm. The average throughput for each algorithm is shown in Table 3.

Table 2: Legend

Original	original backoff algorithm proposed in the IEEE 802.11 standard. Using an exponential increase of *CW*
DCC	the Distributed Contention Control algorithm proposed by L. Bononi 0 where transmission may be deferred depending on slot utilization level during the last backoff.
Konrad	the combined algorithm proposed in this paper. The CW size is computed from the collision average and utilization level during the last backoff.

Table 3: average throughput comparisons

Method	Average throughput (KB/sec)	Performance increase compared to Original (%)
Original	104	-
DCC	111	6.7
Konrad	137	31.7

In the scenario with 32 STAs trying to transmit as fast as possible, our proposed algorithm increased the throughput by roughly 30% compared to the original binary backoff algorithm specified by IEEE 802.11. This 30% increase in throughput is substantial for wireless networks due to their much lower bandwidth and higher protocol overheads compared to wired networks.

Table 3 shows the average throughput of one scenario in a congested system. To get a view of how the different algorithms works under different loads, we ran each algorithm for 4 to 50 STAs in the system observing the average throughput during 60 seconds for each case. The results can be seen in Fig. 4.

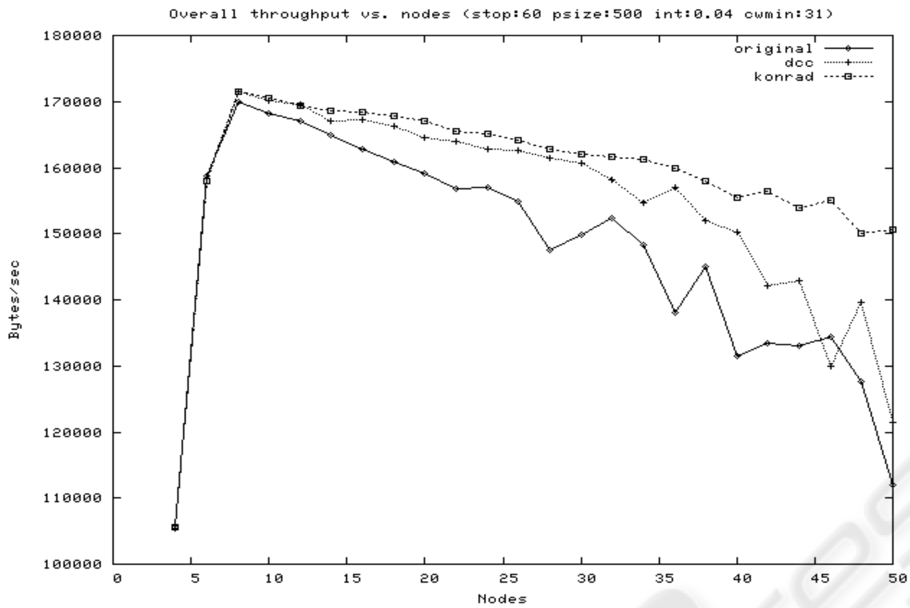

Figure 4: Overall throughput versus number of STAs in the system

The three algorithms perform similarly when the requested bandwidth is less than what the medium can provide. However, when the requested bandwidth rises above that level, the difference between the algorithms becomes visible. It is at this level that the peak throughput will occur, because heavier usage will introduce more overhead in the form of backoffs and collisions, and less usage leaving the medium underutilized. The peak level in our simulations can be seen around 8 nodes in Fig. 4. The algorithm proposed in this paper keep outperforming the other algorithms even when the number of STAs in the system increases up to 50. This feature makes the algorithm ideal to be deployed in systems with a large number of active STAs within one BSS, such as dense office buildings and public access spots.

Collisions Analysis

Another way of measuring how efficient a contention algorithm is to observe the number of collisions generated on the medium. A lower number of collisions means that the algorithm distributes the available bandwidth better between the STAs. Each collision wastes bandwidth and increases the mean delay for the data, which impacts negatively on the throughput of the system. Fig. 5 shows the collision average over time in a system with 32 STAs. All three algorithms have an initial transient peak of collisions before they stabilize, seen at the first 2-3 seconds in Fig. 5. That is because when the algorithms start they do not know the contention level of the medium until they have gained some feedback in the form of collisions and/or slot utilization. Once they have stabilized the collision rate it can be seen that the binary backoff algorithm has the highest collision rate, explaining the lower throughput observed in Fig. 3.

It is also interesting to see the collision rate for different medium loads. Fig. 6 shows the collisions average for the three algorithms with 4 to 50 STAs in the BSS. The collision average is calculated between 5 seconds after start and to the end in 60 seconds. The jump of 5 seconds in the beginning is to avoid the initial transient collision rate seen in Fig. 5.

The fast increase of the collision rate for the binary backoff algorithm seen in Fig. 6 is due to that the algorithm lacks of ability to remember contention levels between successfully transmitted frames. The binary backoff algorithm needs to experience a certain number of collisions before it has adapted its contention window (CW) size appropriately for each new frame to transmit. The binary backoff algorithm has to experience a lot of collisions if the contention level is high, causing a high average collision rate. The collision rate average for the DCC and Konrad algorithms increases in a similar slow fashion with a slightly lower favorable rate for Konrad, making it utilize the medium better.

Now it may look strange that even though the DCC and Konrad algorithms have a similar collision average, the Konrad algorithm gives an additional 20% increase in throughput compared to DCC as seen in Fig. 6. The reason for the difference in throughput despite similar collision rate is because that the DCC algorithm underutilizes the medium while the Konrad algorithm uses the collisions in better way for upcoming transmissions. Because of the way that the DCC algorithm works, it generates collision notification for the backoff scheme not only for missing ACKs but also for when the slot utilization was high during the last backoff time. However, such collision notification based on slot utilization is not based on real collisions on the medium but rather virtual collisions within the algorithm. Because the virtual collision is triggered by a probability based on the fast changing slot utilization value, it may falsely tell the backoff scheme to backoff, hence causing the underutilization of the medium.

Selecting optimal *K*

How the CW depends on the collision average (colAvg) for different values on K is shown in Table 4. Slot utilization equal to zero means that the medium was not accessed at all during the last backoff time, indicating an idle medium. Slot utilization of 0.5 means that half of the backoff slots were busy and 1.0 means that all of the backoff slots were busy.

Table 4: CW size for different slot utilization and K

K	**CW size**		
	SlotUtilization =		
	0.0	0.5	1.0
- 1.0	0	colAvg/2	colAvg
- 0.5	colAvg/2	colAvg	colAvg * 1.5
0.0	colAvg	colAvg * 1.5	colAvg * 2.0
1.0	colAvg * 2.0	colAvg * 2.5	colAvg * 3.0

For a *K* equal to –1.0 the CW size can only be decreased, which gives a maximum CW size of colAvg when the slot utilization is 1.0 (100% of the slots where busy last backoff time). A higher K gives the CW size a higher base value even if the slot utilization is low. The CW size cannot be decreased under colAvg if K equals 0.0 or higher, removing the ability to temporarily decrease the CW size even if the slot utilization value states that the medium is underutilized. To find the optimal *K*, a series of simulations was performed as shown in Fig. 7. The peak throughput is just below a *K* of –0.5 in Fig. 7. The simulation results have illustrated the role of adding the parameter *K* in the Eq. 3. The parameter *K* can give low slot utilization a chance

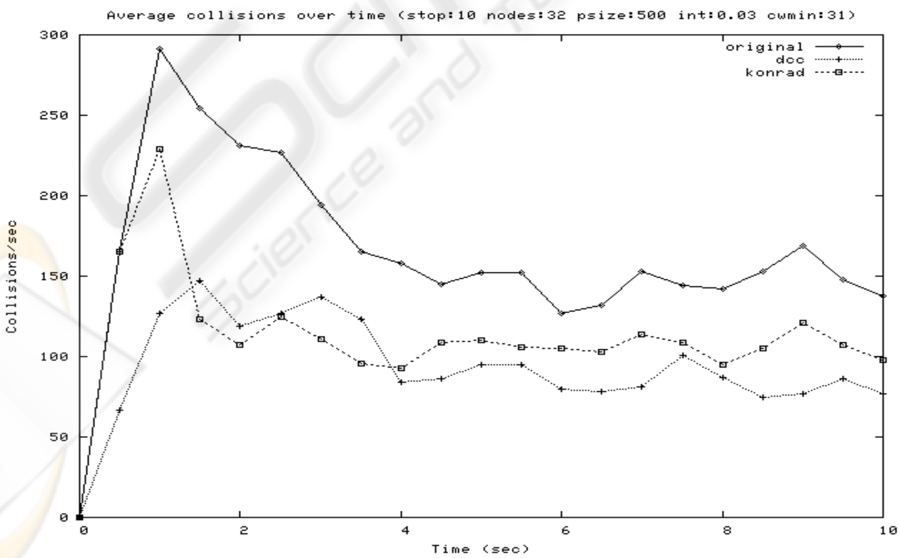

Figure 5: Collision average over time in a system with 32 STAs

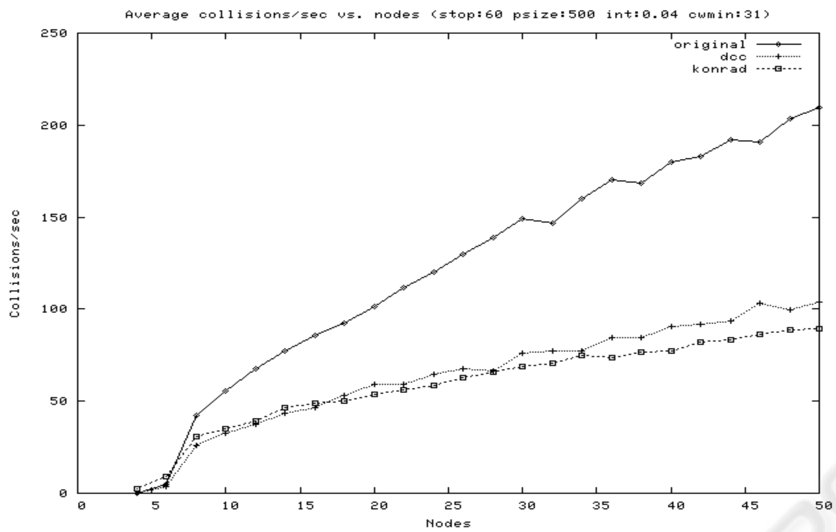

Figure 6: Collision average for different number of STAs STAs

to decrease the CW size, which renders the algorithm able to adapt fast on both increased and decreased contention levels.

Table 4 shows that a *K* of –0.5 moves the offset of the slot utilization as expressed in in Eq.3. So a slot utilization value below 0.5 decreases the CW size while a slot utilization above 0.5 increases the CW size. Slot utilization below 0.5 can be seen as an indication that the medium is not fully utilized because less than half of the slots were busy and a adjustment of the CW size to a lower value would benefit the throughput by giving the STA a smaller range to select its next backoff time from.

The shorter backoff time selected at the next backoff brings the competing STAs retransmission attempts closer to each other, minimizing the mean data delay and maximizing the medium utilization. The increased medium utilization will result in the slot utilization value calculated next time being higher thus not decreasing the CW any further. This feedback behavior makes the algorithm converge to a CW size that creates the maximum utilization.

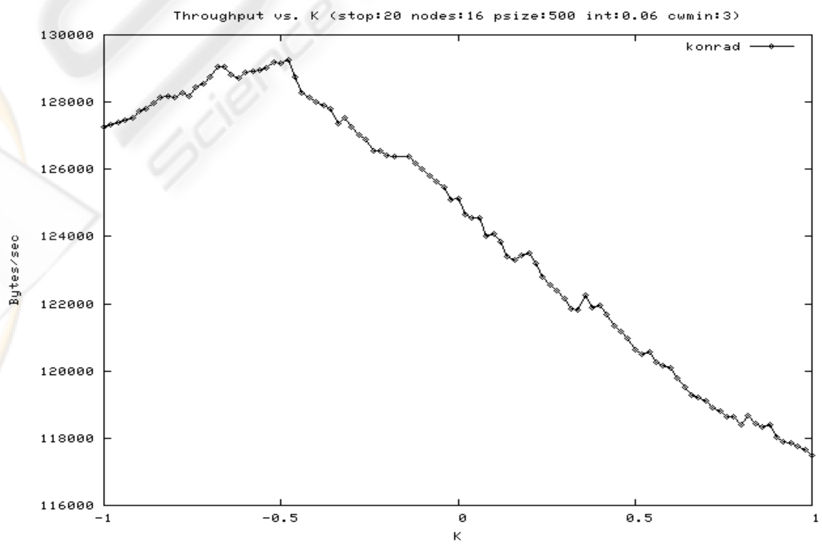

Figure 7: Throughput for different values on *K*

4 CONCLUSIONS

In this paper a new MAC access protocol was proposed. The double feedback introduced can combine transient medium traffic activities and long-term stationary traffic statistics. Extensive simulation has been conducted to compare this new algorithm with the popular binary backoff algorithm proposed by IEEE 802.11 and the DCC algorithm introduced by L. Bononi 0. The throughput gain of our new algorithm was around 20-30% compared to the other algorithms and the gain is increasing when the number of nodes increases. It will be interesting to see how the proposed algorithm affects higher-level protocols like TCP protocols. This is our future research topic.

REFERENCES

Anastasi, G., Borgia, E., Conti, M. & Gregori, E., 2003. IEEE 802.11 ad hoc networks: performance measurements. *Proceedings for the 23rd International Conference on Distributed Computing Systems, pp758-763.*

Bianchi, G., Fratta, L., & Oliveri, M., 1996. Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. *Seventh IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'96) Volume 2, pp392-396.*

Bononi, L., Conti, M.& Donatiello, L., 1998. Design and Performance of a Distributed Contention Control (DCC) Mechanism for IEEE 802.11 Wireless Local Area Networks. *Proceedings of the 1st ACM international workshop on Wireless mobile multimedia, Oct 1998, pp59-67.*

Bruno, R., Conti, M. & Gregori, E., 2002. IEEE 802.11 optimal performances: RTS/CTS mechanism vs. basic access. *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications Volume 4, pp1747-1751.*

Calì, F., Conti, M. & Gregori, E., 2000. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Transactions on Networking (TON) Volume 8 Issue 6, pp785-799.*

Calì, F., Conti, M. & Gregori, E., 1998. IEEE 802.11 wireless LAN: capacity analysis and protocol enhancement. *IEEE Proceedings Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '98) Volume 1, pp142-149.*

Calì, F., Conti, M. & Gregori, E., 2000. IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism. *IEEE Journal on Selected Areas in Communications Volume 18 Issue 9, pp1774-1786.*

Fall, K., & Varadhan, K. The NS-2 Manual (formerly known as *ns Notes and Documentation*), *http://www.isi.edu/nsnam/ns/ns-documentation.html*

Goodman, J., Greenberg, A.G., Madras, N. & March, P. 1998. Stability of Binary Exponential Backoff. *Journal of the ACM Volume 35, pp579-602.*

Kim, H.,& Hou, J. C. 2003. Improving protocol capacity with model-based frame scheduling in IEEE 802.11-operated WLANs. *Proceedings of the 9th annual international conference on Mobile computing and networking, pp190-204.*

Natkaniec, M. & Pach, A.R., 2000. An analysis of the influence of the threshold parameter on the IEEE 802.11 network performance. *IEEE Wireless Communications and Networking Conference Volume 2, pp819-823.*

Natkaniec, M., & Pach, A.R.,2000. An analysis of the backoff mechanism used in IEEE 802.11 networks. *Fifth IEEE Symposium on Computers and Communications, pp444-449.*

P802.11, IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Nov 1997.

Weinmiller, J. & Woesner, H. 1996. Analyzing and Tuning the Distributed Coordination Function in the IEEE 802.11 DFWMAC Draft Standard (1996). *Proceedings of the 4th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS '96).*

Weinmiller, J., Woesner, H. & Wolisz, A., 1996. Analyzing and Improving the IEEE 802.11-MAC Protocol for Wireless LANs. *Proceedings of the 4th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS '96).*

Weinmiller, J., Woesner, H., Ebert, J.P., & Wolisz, A. 1995. Analyzing the RTS/CTS Mechanism in the DFWMAC Media Access Protocol for Wireless LANs. *TC6 Workshop for Personal Wireless Communication (IFIP).*