A Neighborhood Awareness Method for Handoff Assistance in 802.11 Wireless Networks

Gurpal Singh [*], Ajay Pal Singh Atwal[**] and B.S. Sohi[***]

[*] Deptt of CSE & IT, BBSBEC, Fatehgarh Sahib, Punjab, India, gurpal@bbsbec.org
[**] Deptt of CSE & IT, BBSBEC, Fatehgarh Sahib, Punjab, India, ajaypal@bbsbec.org
[***] University Institute of Engineering and Technology, PU, Chandigarh, Punjab INDIA, bssohi@yahoo.com

ABSTRACT

Fast and efficient handoff is one of the foremost requirements of the 802.11 based wireless networks due to their limited range. Prior knowledge of the neighbouring Access points (AP) can assist a mobile station (STA) in making a fast and accurate handoff decision. Dissemination of the neighbouring AP information to all the STA attached to a given AP is a challenging task and should be accomplished without making any changes to the 802.11 standards or huge client/AP modifications or computational complexity keeping in view the backward compatibility issues. We have proposed a new method of providing neighbouring AP channel information to the STA. STA can use this information in conjunction with the syncscan[2] to achieve faster hand off. This approach promises to provide lower signalling overheads in comparison to the traditional syncscan and is also backwardly compatible with the conventional handoff approaches.

Keywords: IEEE 802.11 handoff, neighbour graph, syncscan.

1 INTRODUCTION

IEEE 802.11 based wireless and mobile networks[1] are experiencing a very fast rate of growth and are being widely deployed for providing variety of services but they suffer from limited coverage range of AP, resulting in frequent handoffs, even in moderate mobility scenarios. 802.11 standards follow the "break before Make" approach at the handoff times i.e. connection with the current AP is broken before a suitable AP is searched for making a new connection. It is quite evident from [2][4][6] that the time taken by the STA to scan the neighbouring AP at the handoff time is a major contributor towards the handoff delay in 802.11 based wireless Networks. Long handoff delays are not tolerable in case, real time services are running over the wireless. This time can be decreased if the STA is either provided with the prior knowledge of the channel to which it should shift to in case of handoff, or number of channels to be scanned at handoff time are reduced.

Literature review in this field reveals that researchers have followed either of these approaches to accomplish fast handoff. Hector Velayos and Karlsson[4] have suggested the improvement in detection and scanning phase. In this approach authors have suggested the active scanning mechanism, even with this approach all the channels have to be scanned. Authors have further proposed that to reduce the number of channels to be scanned, neighbour information could be sent as an additional field in beacon frame. Shin et al[6] have proposed a scheme based on neighbour graphs but, this approach is not suitable for fast changing topologies and require long time to build neighbour graphs.. Moreover, all these techniques require significant changes to the 802.11 protocol itself.

In an another approach, Stefan savage and Ishwar Ramani[2] have suggested syncscan mechanism to reduce the handoff delay. In this approach authors have staggered in time, the beacons generated by the AP working on different channels. In this manner STA are made aware of the expected beacon arrival times of other channels. Using this information, STA can listen to beacons from other channels by pretending to the current AP of going into power save mode. This approach replaces the transient scanning overhead with a regular scanning overhead. This is an elegant scheme but STA still have to run syncscan on all the channels as they have no method of knowing about their neighbours. Moreover this approach adds to high signalling overhead if syncscan interval is reduced[2], also the usefulness of syncscan is not optimal, in case of bulk data transfer[2].

In this paper we have proposed a scheme to disseminate accurate neighbouring AP channels information. This is accomplished while the STA is still connected to the current AP and without making modifications into the existing 802.11 standards. We have done analysis of this approach and have calculated the percentage signalling overheads due to this approach. This scheme provides excellent results if used in conjunction with the syncscan approach.

2 BASIC ALGORITHM

Our algorithm proposes that all APs in the Distribution system (DS) know about the channel numbers on which their immediate neighbouring APs are operating. This Neighbour information is further disseminated to the STA's, by the current AP to which they are attached.

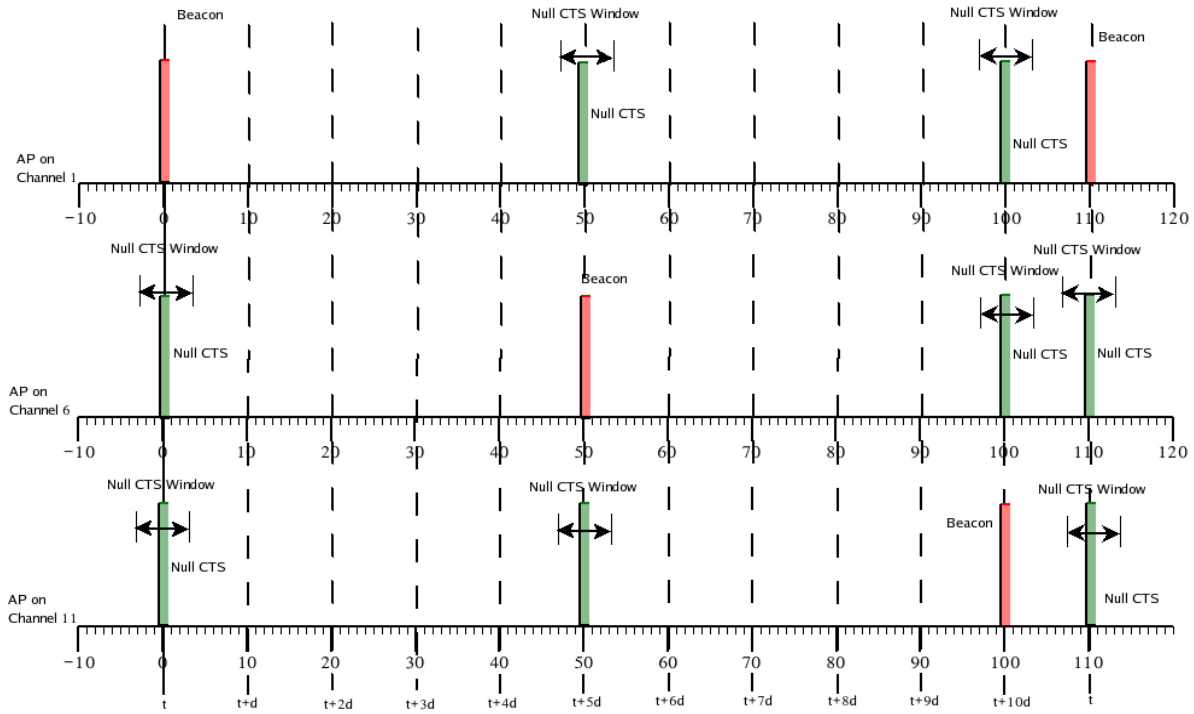


Fig 1: Diagram shows scheduling of Beacon frames and null CTS for three non overlapping channels with beacon interval of 100ms.

We further propose that all the APs are synchronised and their beacon timings are staggered in time as in the case of syncscan algorithm. Some approaches to gather neighbour information at the APs are mentioned below.

A simpler approach can be to manually build a neighbour information table named APINFO, consisting of neighbouring AP channel numbers at the time of installation of DS. For a smaller DS manual this approach can be quite cost effective. For synchronisation purposes Network Time Protocol (NTP) can be used.

Another approach is to GPS enable all APs[3]. GPS will serve two purposes, firstly, it will provide the timing synchronization to the AP's which otherwise has to be provided with the help of NTP. Secondly, GPS will provide the position coordinates of the APs. Whenever an AP is switched on it send it's GPS coordinates on the DS with the help of modification suggested in the IAPP protocol[5]. In response to this all the AP's on the DS send back their GPS coordinates and channel numbers on which they are working. The new AP will calculate it's geographical neighbours and also assign a non overlapping channel number to itself. All this information between the APs, is exchanged with the help of IAPP protocols after regular intervals of time [5].

Another approach is to build the neighbour information using the information in Reassociation Requests sent by STA at the handoff times. The information about the neighbours and their channels number is stored by the AP's in the form of an APINFO table.

The task of disseminating this neighbour information to the STA attached to a given AP is a complex task and it needs to done with desirably no modifications to the existing standards, minimum modifications to client/AP software and minimum signalling overheads. To accomplish this task we exploit the mechanism used by syncscan, and use it for propagating neighbour information to associated STAs.

In syncscan algorithm beacons from the AP's working on different channels are staggered in time. We further extended this approach to provide neighbour information to the STA we call this approach as Sync CTS.

3 SYNC CTS

As described in the previous sections, all STA attached to a given AP are aware of the exact time of arrival of beacons from the other channels[2]. It is proposed that an AP sends specially crafted null CTS frames with duration field set to zero and destination address of self, at the exact times when beacons from one of the neighbouring AP, i.e. channels contained in the APINFO tables, is expected. Based on this, if a null CTS is received by the STA at the exact moment when the beacon from a neighbouring channel was expected , it is taken as an indication of presence of that channel. Likewise, STA can gather information about all the neighbouring AP channels. It should be noted that Null CTS transmission can be delayed in case, media is not available due to some ongoing transmission. We have circumvented this problem by proposing a Null CTS window during which an AP can send a null CTS frame. If Null CTS is received during this Null CTS window time, it is taken an indication of presence of a channel. By doing so the stringent timing requirements are taken care of.

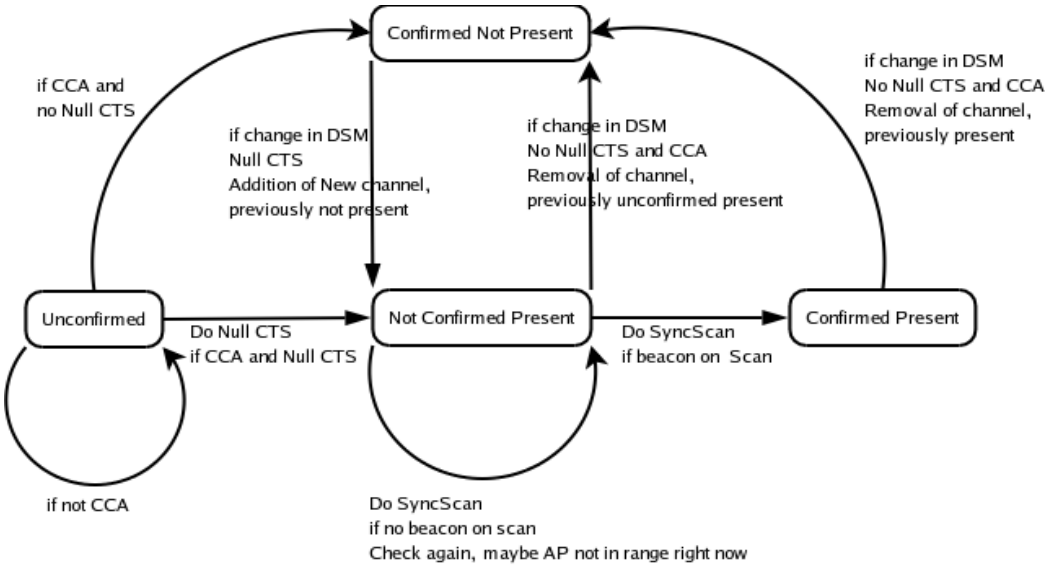


Fig 2: State Machine Depicting How channels are selected for syncscan

We propose null CTS frames with zero duration and receiver address equal to sending AP address. Such null CTS frames are sent by the AP only at the times when beacons from the neighbouring APs are expected as shown in Fig 1.

All the null CTS for a given AP are sent during a beacon interval and occurrence of such beacon intervals takes place only either at the time of associations/reassociation or in case there is some change in the APINFO table. To ensure that neighbour information properly reach the STAs, the null CTS frames are repeated over the next beacon interval as well. This is to ensure that if some STA is undergoing syncscan operation it can listen to null CTS frames in the next beacon interval. Thus, the list of neighbouring channels is gathered by the STA. The list of neighbouring channels gathered so is fed to the syncscan algorithm running on the STA. Thus, Syncscan algorithm will run on limited number of channels as opposed to all the channels. We have defined the states into which a system can move and the State transition diagram is given in Fig 2.

- **Unconfirmed(UC)**: Initially all the 802.11 channels are in this state. We use classic syncscan if required.
- **Not confirmed Present (NCP)** : This state contains the channels for which null CTS was received from the currently associated AP. This state contains the list of neighbouring channels on which syncscan algorithm will actually run to ascertain their signal strengths.
- **Confirmed Present (CP)**: This state contains the list of channels on which syncscan has successfully received the beacons and measured their signal strengths. These channels will be used for making handoff decisions.
- **Confirmed Not Present (CNP)**: This state will contain the list of channels for which channel was sensed to be clear at the time when null CTS was expected but no null CTS was received. Syncscan will never run on these channels.

Initially when a new STA enters the DS it listens for at least two beacons and synchronizes itself with the AP[2]. At this stage all the channels are in the UC state. Immediately after grant of association to the new STA by the AP, the AP sends a null CTS, at the times when beacons are expected from one of the neighbouring channel. This is repeated over the next interval as well, for each neighbour. Neighbour list is taken from the APINFO table of the AP. The neighbouring channels for which Null CTS are received are placed into UCP list and syncscan is performed only on these channels, after performing syncscan are put into CP state or remain in UCP. All the handoff decisions by the STA are made based on the signal strengths of the channels in the CP state.

3.1 Comparison with the classical syncscan approach

In the proposed approach all STA are provided with list of neighbouring channels which are operating in the vicinity of a given AP. Lesser number of channels means that syncscan operation has to performed lesser number of times. Classical syncscan is shown in Fig 3(a). It is quite evident that in our approach syncscan has to run less frequently and this results in significant saving in the syncscan overheads. This can be further classified into two categories:

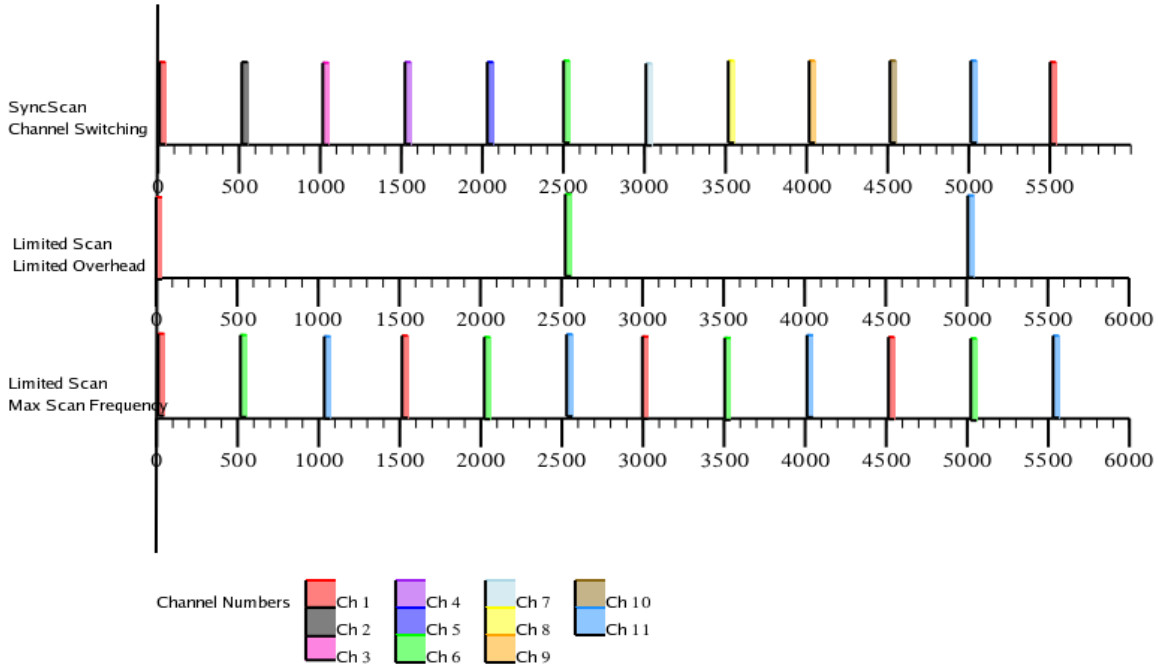


Fig 3: Null CTS frequency versus scanning overheads with syncscan interval of 500ms with 3 neighbors. (a) Classic SyncScan, (b) Low mobility scenario, (c) High mobility scenario respectively.

Low Mobility Scenarios: When STA mobility will be low and consequently, the need of refreshing the signal strength value for a given neighbour will be lesser. For a refresh rate comparable to that of classical syncscan, our approach requires far less number of syncscan operations. This will result in reduced signalling overheads and the probability of packets being buffered at the AP will also decrease and consequently the average packet jitter will also get reduced. This is shown in Fig 3(b).

High Mobility Scenarios: When STA mobility will be high, faster neighbour signal strength refresh rate will be required. By keeping the syncscan interval same as that of conventional syncscan, our approach will provide far better neighbour signal strength refresh rate which is a desirable feature in the high mobility scenarios. In this case signalling overheads and jitter will remain same but signal strength update frequency for a given neighbour will increase. This approach will be suitable for the scenarios when user mobility is high and signal strength of the prospective neighbouring AP s has to be monitored at a faster rate. This is shown in Fig 3(c).

3.2 Overhead due to Null CTS

In this section we calculate the overhead traffic due to the Null CTS frames in a given beacon interval and is represented in the form of percentage overhead of the normal traffic. To calculate this we assume that STA has just associated with a given AP. As proposed in the algorithm the AP will send Null CTS frames in the beacon interval that follows the re association request. In this case two calculations are being performed. The first calculation is for finding out the normal throughput in a contention based environment in absence of Null CTS frames. In Second calculation, overheads due to Null CTS are calculated.

3.2.1 Calculation of Normal Throughput

For calculating the normal throughput we have adopted the model proposed by [8].

In a contention based environment as per [8] the probability *p*, of traffic of a given STA colliding with transmission of any of the other STAs can be approximated as:

$$p = 1 - \left[1 - \frac{2*(1-2p)}{(1-p-p(2p)^m)} \cdot \left(\frac{1}{W}\right)\right]^{(n-1)} \qquad (1)$$

Where

n is the number of STAs

W is the minimum window size

m is the back off stage and max windows size is 2^m*W

T_{slot} is slot time

$T_{payload}$ is time to transmit payload bits

T_{phy} is time to transmit packet

T_{cycle} is time between start of two packet transmissions.

Equation (1) can be solved for *p* by simplifying and applying Newton bisection method for various values of n, W and m. Table 1 shows the values of p for various values of n, with constant W and m.

Due to contention based access mechanism in 802.11, certain portion of total transmission results in collisions. From (1) Success rate of transmission can be calculated which is given in equation (2)[8].

n	*W*	*m*	*p*
5	128	3	0.0625
10	128	3	0.119
15	128	3	0.168
20	128	3	0.231

Table: 1

$$r_{success} = \left(\frac{2*(1-p)}{(2-p)}\right) * \left(\frac{1}{T_{cycle}}\right) \quad (2)$$

Where

$$T_{cycle} = T_{physical} + T_{SIFS} + T_{ACK} + T_{DIFS} \quad (3)$$

It is assumed that all packets are of uniform size and all the STAs are in saturated stage, i.e. they always have a packet to transmit.

In this case

packet payload	8184 bits
MAC Header	272 bits
PHY Header	128 bits
ACK Length	240 bits
Channel Bit Rate	1 Mbits/ sec
SIFS	28 microseconds
DIFS	130 microseconds
SLOT Time	51 microseconds
Channel Bit Rate	1 Mbits/sec
propagation delay	1 microseconds

The throughput is given by:

$$Throughput = R_{success} * PayloadSize \quad (4)$$

$$Throughput = 2\frac{(1-p)}{(2-p)} * \left(\frac{PayloadSize}{T_{cycle}}\right) \quad (5)$$

Where units of T_{cycle} are taken in seconds and payload size is taken in bits and hence throughput is measured in bits per second. Now the throughput in a single beacon interval of 100ms duration is obtained as follows:

$$Throughput_{beaconInterval} = \frac{Throughput}{10} \quad (6)$$

3.2.2 Traffic due to Null CTS

It has been proposed in our algorithm that null CTS are sent during the times when beacons are expected from neighbouring channels. We have defined a window named Null CTS Window during which if a Null CTS frame is received, it is considered as a signal for the presence of the neighbour. The provision of Null CTS window is to provide certain degree of tolerance in time, as the media may be busy at the precise time when the beacon from an AP on a neighbouring channel is expected. The size of Null CTS window can be varied and moreover, it does not have any effect on existing traffic. In case the AP is expecting an ACK for data sent, it can delay the Null CTS frame for SIFS interval after the ACK, so as to avoid collision with ACK which has same priority as Null CTS. The transmission of Null CTS frame is contention free (highest priority) and therefore there is no need of finding the probability of collision of a Null CTS frames with traffic from other STAs. The only condition is that for successful transmission of null CTS there should be at least one transmission opportunity in the Null CTS window.

Let us assume that $T_{nullctswindow}$ is the size of Null CTS window and T_{cycle} represents the time to transmit a given packet. T_{cycle}, therefore represents the interval between the two packet transmissions. All the above intervals are taken in slot times.

Let

$$x = \left\lceil \frac{T_{nullCTSwindow}}{T_{cycle}} \right\rceil \quad (7)$$

represents the number of transmissions in a given null CTS window. Null CTS window size is always taken larger than the average packet transmission time so that there is at least one transmission opportunity in a given null CTS window. Also Null CTS window should not be too large which can increase the chances of Null CTS window overlapping or false detection of neighbours. In our calculations we take x=1 for

$$T_{Nullctswindow} \geq T_{cycle}$$

AP will grab any transmission opportunity during Null CTS window to send Null CTS frame as it has highest priority(Null CTS Transmit after waiting for SIFS) and does not have to content for the media. There is also no chance of Null CTS frames collision with transmission from other STAs as it is assumed that there are no hidden terminals.

$$Null\ CTS\ overhead = x * NullCTSlength * Neigbours \quad (8)$$

Where size of null CTS MAC is 14 bytes and it contains the address of the AP which is sending it. Duration field in such a Null CTS frame is set to zero. AP sends a Null CTS frame addressed to self and also transmits it on the radio interface so that it is heard by all the STAs. Time taken to send a Null CTS frame is given below. This overhead is actually calculated on the basis of Null CTS frames sent in the beacon interval just succeeding the association/ reassociation request. It is expressed in number of bits used to send null CTS frames corresponding to all the neighbours per beacon interval.

Length of Null CTS = SIFS + Physical header + (Null CTS MAC)

Where Size of null CTS MAC=14x8 bits

Physical Header size= 128 bits.

Number of neighbour depends on the deployment and available in APINFO table stored on AP.

Percentage overhead:

$$overhead = \left[\frac{(Null\ CTS\ overhead)}{Throughput_{baeconinterval}} \right] * 100 \qquad (9)$$

This percentage overhead calculation is applicable only for the beacon interval carrying the NULL CTS signalling frames. In all other beacon intervals, the normal traffic persists. Graph 1 shows the percentage overhead versus number of neighbours for various number of STAs attached to a given AP.

3.3 Backward compatibility

Our approach is fully backward compatible with no modifications done to the base 802.11 standards. In case STA is not running the modified drivers it will simply ignore the Null CTS signalling frames and can use the classical syncscan or the conventional 802.11 method for handoff decision.

4 CONCLUSION

In this paper we have proposed a novel approach of providing neighbour information to the 802.11 wireless STA. This approach makes use of synchronized null CTS frames for disseminating neighbour information. Null CTS are very small frames (14 bytes) and add very little to the signalling overheads. Null CTS are sent only at the times, when beacons from the neighbouring channels are expected. Priority of CTS frames is highest as they are sent after SIFS and consequently the probability of sending a null CTS frame at the precise times is very high. By providing neighbour information, number of channels to be scanned by the STA is reduced resulting in lesser number of syncscan operations. This approach is fully backward compatible and can be modified to suit both high mobility and low mobility scenarios. We have mathematically calculated the percentage overheads for the beacon intervals carrying the null CTS signalling elements and the results show that the overheads are within the permissible limits. Moreover, Null CTS signalling overheads are not continuous and are present only at the time of association/ reassociation requests. We are currently investigating this approach on our experimental test bed to substantiate the theoretical findings.

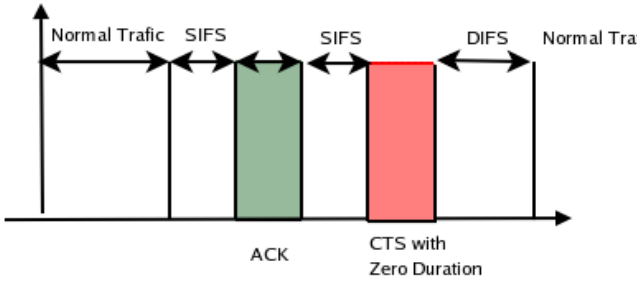
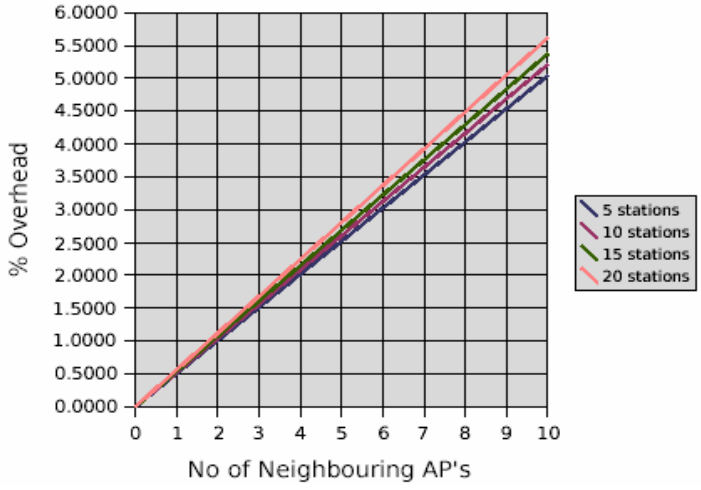


Fig 4: NULL CTS Transmission



Graph 1: Overhead due to NULL CTS with varying number of neighboring AP's

REFERENCES

[1] IEEE. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification. IEEE Standard 802.11, 1999.

[2] I. Ramani, S. Savage, "syncscan :Practical Fast handoff for 802.11 Infrastructure Networks," *IEEE INFOCOM* 2005.

[3] GPS Guide for beginners, http://www.garmin.com/manuals/GPSGuideforBeginners_Manual.pdf".

[4] H. Velayos and G. Karlsson, "Techniques to reduce IEEE 802.11b mac layer handover time," Kungl Tekniska Hogskolen, Stockholm, Sweden, Tech. Rep. TRITA-IMIT-LCN R 03:02, ISSN 1651-7717, ISRN KTH/IMIT/LCN/R-03/02–SE, April 2003.

[5] IEEE. Recommended Practice for Multi-Vendor AP Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation. IEEE Draft 802.1f/D3, January 2002.

[6] M. Shin, A. Mishra, and W. Arbaugh, "Improving the latency of 802.11 hand-offs using neighbor graphs," *in Proceedings of the ACM MobiSys Conference*, Boston, MA, June 2004.

[7]Gurpal Singh, Atwal Ajay Pal Singh, Rajbahadur Singh, BS Sohi, IAPP modifications for a location based handoff technique in wireless Networks, SOFTWIM 2006, Proc of Workshop on Software for Wireless Communications and Applications Co-located with COMSWARE 2006, New Delhi, India, 2006.

[8] Tay Y.C and Chua K.C., "A Capacity Analysis for the IEEE 802.11 MAC Protocol" *in Wireless Networks vol 7 p159-171, Kluver Academic Publishers, 2001.*