Overlapped NACKs: Improving Multicast Performance in Multi-access Wireless Networks

Luca Canzian, Andrea Zanella and Michele Zorzi
Department of Information Engineering, University of Padova, Via Gradenigo 6/B, I-35131 Padova, Italy
canzian@dei.unipd.it, zanella@dei.unipd.it, zorzi@dei.unipd.it

Abstract—A wide variety of network applications require the use of reliable multicast protocols to disseminate data from one source to a potentially large number of receivers simultaneously. We focus on multicast in single channel multi-access wireless networks and consider Automatic Repeat reQuest (ARQ) based error control algorithms. We propose a simple NACK based method, called overlapped NACKs, that exploits the physical nature of the transmitted NACK signals in order to achieve a high Signal to Noise Ratio (SNR) even in the presence of NACK collisions. As an example, we apply the proposed method to Bluetooth, quantify its reliability and compare the achievable throughput with respect to the multi-unicast case and to an ACK based protocol. Results show that the overlapped NACK method has the potential to substantially increase the throughput over the ACK based protocol and provides good reliability.

I. INTRODUCTION

Multicast is an efficient paradigm to transmit data to a group of receivers, which brings lower network costs and bandwidth consumption than multiple unicast to individual group members. A wide variety of applications can benefit from multicasting, including multimedia conferencing, distance learning, multi-party games, software and database updates. In particular, delay-sensitive applications are becoming more and more popular. For these types of applications, usually no error control algorithms are implemented during a multicast communication, because they would significantly degrade the performance of the system.

In this paper we present an error control algorithm that provides a fairly good reliability while being very efficient in terms of throughput performance. We focus on multicast in single channel multi-access wireless networks and consider Automatic Repeat reQuest (ARQ) based error control algorithms. According to a classic ARQ mechanism, a small acknowledgement (ACK) packet is sent by the receiver to the transmitter to inform that a correct data frame is received. If no ACK is received, the corresponding data frame is retransmitted. In a single channel multi-access scenario this basic mechanism requires strong coordination among nodes to avoid packet collisions. Moreover, an increase in the number of nodes leads to higher bandwidth requirements and to the well known feedback implosion problem, which refers to the significant overhead incurred by the transmitter to process the ACK sent by each receiver. A potential solution to this problem [1] is obtained by shifting the burden of providing reliable data transfer to the receiver, using a negative acknowledgement (NACK) packet to inform the transmitter of an erroneous packet reception. If a NACK packet is received, the corresponding data frame is retransmitted. Only receivers detecting an error or a packet loss send feedback to the sender. However, even with a NACK based protocol, NACK bandwidth consumption and NACK implosion may still pose serious scalability problems. For this reason, methods to suppress generated NACKs have been considered. In [2] and [3] a delay based protocol (DBP) is analyzed, based on NACK multicasting and random delays prior to NACK generation. In [4] a leader based protocol (LBP) is proposed. It is a mixed approach where a receiver is elected leader and only this leader is allowed to send ACKs, while other receivers can only transmit NACKs that may collide with the ACK sent by the leader, causing a retransmission of the data packet. Analytical results suggest that LBP performs better than DBP. However, LBP itself has a major drawback when applied to mobile nodes. A new leader will have to be selected each time the current one leaves the network. As mobility increases, this problem becomes prominent. In [5] tones are used, instead of conventional packets, to signal a NACK. Dual tones, proposed by [6], are also incorporated to prevent an incoming mobile node from interrupting an ongoing transmission. This approach requires two antennas, one for busy tones and the other for data packets and NACK tones. In [7] a carrier sense ARQ scheme has been studied. This approach can be exploited to manage collision of NACK packets sent by different receivers.

All these methods assume destructive collisions. Conversely, our work is based on the observation that, even in the presence of packets collision, the Signal to Noise Ratio (SNR) of the received signal may still be higher than a threshold, thereby allowing its correct recognition. The proposed method, that we call overlapping NACKs, is a NACK based protocol where NACK packets sent by receivers are identical. If a NACK collision occurs, the transmitter receives a signal that results from the overlap of equal signals with different powers and phases. Therefore the resulting signal is still of the same form, but its SNR depends on the amplitude and relative phases of the signals involved. The trick is to minimize the probability of destructive NACK overlapping. This approach is more robust to interference than other similar methods, such as those proposed in [5] [6] [7], that are more prone to false alarm, especially in a noisy scenario, causing a significant number of useless retransmissions with a consequent degradation in terms of bandwidth allocation, power consumption and throughput.

In this paper, we describe the overlapping NACKs protocol, a very simple method able to exploit the native tools provided.

1Throughout the paper, we use the terms packet and frame interchangeably.
by the considered network technology. As an example, we will see how this method can be easily implemented in a common and popular technology like Bluetooth. We quantify the performance, in terms of throughput, and compare it to the multi-unicast strategy supported by Bluetooth and to a positive ACK based protocol that is being considered for the efficient implementation of multicast in Bluetooth. We will see that the proposed method is actually capable of significantly enhancing the system performance.

The remainder of this paper is organized as follows. Section II describes the proposed NACK based method, called overlapping NACKs, starting from the assumption that NACK packets are perfectly synchronized. Then results are generalized introducing a time misalignment between NACKs. In Section III the role of the amplitude parameter is analyzed and a solution to keep its value high is described. In Section IV the proposed method is applied to Bluetooth and compared to the ACK based and multi-unicast protocols, in terms of achievable throughput. Moreover, the detection performance of the overlapped NACKs method is studied. Section V concludes the paper with some final remarks.

II. OVERLAPPED NACKS

To illustrate our approach, we consider the reliable transmission of a packet from a node, that we call master, to a group of nodes, that we call slaves. The master addresses the packet to all slaves simultaneously. Since the proposed method is a NACK based protocol, the master transmits a packet only if it receives a NACK. Each slave sends a NACK immediately after packet transmission if an erroneous packet or no packet (for synchronous links) has been received. So there is no waiting involved as in delayed feedback-based methods, thereby avoiding wasted channel bandwidth and improving the performance. Once the master recognizes a NACK, it retransmits the packet.

The problem we analyze here is to find the particular structure of a NACK packet that allows the master to recognize a NACK in the presence of packet collisions. In order to derive the final structure of the NACK, we first study the case in which NACK packets arrive at the master perfectly synchronized, and then extend the results to the more general asynchronous case. Assuming that a NACK is a standard packet, equal for all slaves (i.e., containing the same sequence of symbols), we can easily realize that the received signal has the same form of the original signals. Let us consider for simplicity the case of a Phase-Shift Keying (PSK) modulation, though these concepts can be easily extended to QAM, FSK and their variants. The NACK received by the master from a generic slave i, assuming an ideal channel (attenuation and delay), is represented by a signal of the following type:

\[ s_i(t) = A_i(t) \cos (2\pi f_c t + \varphi_i(t) + \theta_i) \]  

where \( f_c \) is the carrier frequency, \( \theta_i \) is the initial phase, \( \varphi_i(t) \) is the phase of the transmitted signal,

\[ A_i(t) = A_i \left| \sum_{j=0}^{M-1} p(t - jT) \right| \]  

is the amplitude of the transmitted symbols, \( M \) is the number of transmitted symbol and \( p(t) \) is the transmission pulse. The scale factor \( A_i > 0 \) depends on transmission power and channel attenuation. If \( p(t) \) is rectangular then \( A_i(t) = A_i, \forall t \) and the power of the received signal is equal to \( P_i = A_i^2/2 \). \( P_i \) and \( A_i(t) \) are in general tied by a square relation, therefore maximizing the SNR is equivalent to maximizing \( A_i(t) \). Under the assumption that NACKs are represented by the same packet (i.e., \( \varphi_i(t) = \varphi(t), \forall i \)), it is easy to show that the overlapping of \( N \) signals represented by (1) assumes the following form:

\[ s(t) = \sum_{i=1}^{N} s_i(t) = A_s(t) \cos (2\pi f_c t + \varphi(t) + \theta_s) \]  

where:

\[ A_s(t) = \left| \sum_{i=1}^{N} A_i(t)e^{j\theta_i} \right| = A_s \left| \sum_{j=0}^{M-1} p(t - jT) \right| \]

\[ A_s = \sum_{i=1}^{N} A_i e^{j\theta_i} \]

\[ \theta_s = \left( \sum_{i=1}^{N} A_i(t)e^{j\theta_i} \right) = \sum_{i=1}^{N} A_i e^{j\theta_i} \]  

The received signal (3) has the same structure of (1). A graphical interpretation of this result is given in Fig. 1 where two QPSK constellations, having parameters \( A_1, \theta_1 = 0, A_2 \) and \( \theta_2 \), are summed. The resulting constellation is still QPSK, with parameters \( A_s \) and \( \theta_s \), depending only on the amplitudes and phases of the two original constellations.

A. Time misalignment

Equation (3) has been obtained under the unrealistic hypothesis that NACKs from different slaves all arrive synchronized at
the master. When time misalignment between NACK packets is introduced, we have to find an interval where (3) is still valid. Let $K$ be the smallest integer such that the probability that the time misalignment between two NACK packets is higher than $KT$ is sufficiently small, where $T$ is the symbol period (in other words, $KT$ is the maximum time misalignment). Suppose that each NACK packet is made of $M$ symbols, with $M > K$. Then there is a time interval of length at least $(M - K)T$, that we call overlapping interval, where all NACK packets overlap. In order to assure that NACK packets assume the same form in the overlapping interval, each NACK must be composed by the same symbol repeated over time, i.e., $\varphi_i(t) = \varphi, \forall i, t$. Actually, this constraint is equivalent to $\varphi_i(t) = \varphi, \forall t$ since the difference between $\varphi$ and a reference phase $\varphi$ can be included in the initial phase term $\theta_i$. Therefore a NACK packet assumes the form:

$$s_i(t) = A_i(t) \cos (2\pi f c t + \varphi + \theta_i)$$  (5)

However, the resulting signal still assumes the form (5) in the overlapping interval only if NACKs are misaligned by an integer multiple of the symbol period $T$, or if the transmission pulse $p(t)$ is rectangular. In general, we obtain an amplitude $A_i(t)$ that does not have the form of the transmission pulse $p(t)$, and an initial phase $\theta_i(t)$ which depends on time. However, since $p(t)$ is a finite pulse, (5) represents a periodic signal with period $T$, except for a small interval at the beginning and at the end of the packet (border effect). Therefore there is an interval, that we call recognition interval, where NACKs overlap and are periodic signal. The recognition interval is a subset of the overlapping interval, and they coincide only if there are no border effects in the overlapping interval. If the duration of $p(t)$ is $nT$, then the recognition interval length is at least $(M - K - n + 1)T$. Since the sum of periodic signals is itself a periodic signal, during the recognition interval the received signal, $A_i(t)$ and $\theta_i(t)$ are periodic too. Once the received signal is sampled with a sampling rate $1/T$, both $A_i(t)$ and $\theta_i(t)$ become constant, so that the sampled version of our signal is equivalent to the sampled version of a PSK waveform.

Therefore, we conclude considering the following rule for a NACK recognition: the master must identify at least $N_{sym} = M - K - n + 1$ consecutive equal symbols. This can be done during the recognition interval if the SNR, or alternatively the amplitude $A_i(t)$, is large enough. This is analysed in Section III.

It is important to note that for $K \gg N_{sym}$ (i.e., for technologies where propagation delays or asynchronism between slaves are relevant) the proposed method is not efficient because the NACK should be composed of many more symbols ($M$) than the sequence length that must be recognised ($N_{sym}$). In this case a differential approach can be considered in order keep the method efficient ($M \simeq N_{sym}$). The master can differentiate consecutive received symbols and count the number of differential symbols equal to zero over an interval of $N_{tot}$ symbols. If a new NACK packet arrives during this counting, its effect is to decrease the counting by $n$ units (usually $n \simeq 1$). Therefore it could be sufficient to consider $N_{tot}$ slightly greater than $N_{sym}$ in order to recognize a NACK also during a time interval different from the recognition interval.

Finally, we remark the fact that the proposed method is robust against multi-path since its effect is to add other NACK replicas to the received signal, increasing the number of overlapping NACKs which, statistically, is a positive effect, as we can see in Section III. Besides, the extra misalignment produced by such replicas is usually negligible since they are very close to the original ones. As an example, in Bluetooth the maximum misalignment between two packets sent in the same slot by different transmitters is, from specifications, $20\mu s$. A multi-path replica, in order to produce the same misalignment, would require a path length difference of $2$ km, which is unrealistic in the environments we consider.

III. Amplitude Parameter

While the initial phase parameter $\theta_i$ is not critical because its constant value can be easily estimated and compensated by the master, the amplitude parameter $A_i(t)$ is fundamental in order to prevent wrong phase evaluation due to noise or interference at the receiver, i.e., in order to keep the SNR high. In (4), $A_i(t)$ is proportional to a repetition of $p(t)$ and $A_i$ depends only on $A_i$ and $\theta_i, \forall i$. In general, in the presence of time misalignment, $A_i(t)$ assumes a more complicated form: during the overlapping interval, it is proportional to a sum of repeated and differently time-shifted versions of $p(t)$. However, in many cases of interest (e.g., for rectangular pulses), since the periodic repetition of $p(t)$ is constant, $A_i(t)$ can still be represented by (4). The general form of $A_i(t)$ only requires a more cumbersome notation, without adding any relevant concept.

In order to maximize $A_i = \sum_{i=1}^{N} A_i e^{j\theta_i}$ all signals should be in-phase, i.e $\theta_i = \theta, \forall i$. However, there is no control on the phases $\theta_i$, that depend on the local oscillator of each slave and on the propagation parameters (e.g., distance and reflections) of each NACK signal. We can assume they are uniformly distributed in the interval $[0, 2\pi]$. In order to calculate the probability distribution of $A_i$, we should make some assumptions on the probability distribution of $A_i$. However, under the hypothesis that $A_i, i = 1, \ldots, N$ are independent and equally distributed with $A_i > 0$, it is easy to show that $E[A_i] > E[A_k]$ and $E[A_k]$ is an increasing function of $N$. For this reason, and for space constraints, we do not provide a detailed statistical analysis but, rather, in order to illustrate a simple and efficient algorithm, we briefly consider the specific case where $N = 2$ slaves are transmitting NACK packets that arrive at the master with equal power, i.e., $A_1 = A_2 = A$. Since phases are uniformly distributed it is easy to see that $E[A_k] = \sqrt{2}A$, higher than the amplitude of each single NACK. However, if $\theta_2 - \theta_1 \simeq \pi$ then $A_k \simeq 0$. In this situation the SNR would probably be too low to allow NACK recognition. In order to reduce the missed detection probability, we propose a double NACK scheme: two consecutive NACKs are sent instead of only one. During the first NACK all slaves send an identical symbol, while in the second NACK each slave sends a symbol that is characteristic of the slave. If double NACK signals of slaves 1 and 2 have opposite phases during the first NACK, then in the worst case during the second NACK we obtain
\[\theta_2 - \theta_1 \simeq \pi - 2\pi/M\] if an \(M - \text{PSK}\) modulation is considered. For example, if a \(QPSK\) is considered, during the second NACK we would obtain \(A_s = \sqrt{2}A\). Obviously during the second NACK each slave sends a symbol that is characteristic of the slave only if the number of slaves of the multicast group, \(N_{\text{slaves}}\), is lower than the alphabet size, \(M\). If this is not true, we can generalize the double NACK concept proposing an \(n\)-NACK scheme: \(n\) consecutive NACKs must be sent. In order to assure that there is a relative change of phase between each pair of slaves in at least one of these \(n\) NACKs, it must be \(N_{\text{slaves}} \leq M^{n-1}\) or, equivalently, \(n \geq 1 + \log_M N_{\text{slaves}}\), which shows that the NACK length increases logarithmically with the number of members in the multicast group. Simulation results show that the \(n\)-NACK method significantly reduces the missed detection probabilities, thereby increasing the reliability of the overlapped NACK protocol.

### IV. Case Study: Bluetooth

In this section we first shortly overview the features of the Bluetooth technology that are of interest for our analysis, and then describe two alternative multicast schemes and compare their performance with that of our proposed protocol. Readers interested in the details of Bluetooth are referred to the official standard [8].

The basic Bluetooth network configuration is the piconet, a cluster of no more than eight devices sharing a common frequency-hopping radio channel. When the piconet is established, one device gets the master role, while the others get the slave role. Direct communication can occur only between master and slaves by means of a Time-Division Duplexing (TDD) communication. Master’s transmission commands the recipient slave to return a packet in the following slave’s slot. Each slot lasts 625 \(\mu\)s. In this paper we consider only Asynchronous Connection-Less (ACL) packets which begin with the access code, used for signal acquisition, followed by the header, that carries control information. Access code and header are always transmitted at the basic rate \(R_1 = 1\) Mbps and modulated with a Gaussian Frequency-Shift Keying (GFSK) modulation. The upper layer data is carried in the payload field, which can be transmitted at \(R_1 = 1\) Mbps, \(R_2 = 2\) Mbps or \(R_3 = 3\) Mbps. Time occupancy of ACL frames is limited to 1, 3 or 5 consecutive time slots. Different packet types will be denoted by \(jDHn\), where \(j = 1, 2, 3\) is the transmission rate in Mbps, while \(n = 1, 3, 5\) is the slot occupancy. In a reliable communication, packets are retransmitted until the sender gets an ACK, piggybacked by the ARQN flag in the header of the return packet from the destination.

#### A. Multicast Proposals

Bluetooth realizes a reliable point to multipoint communication from the master to more slaves, sending the same packet to each slave separately, i.e., using a multi-unicast communication. This mechanism requires a lot of power and bandwidth. For this reason the Special Interest Group (SIG), the body that oversees the development of Bluetooth standards, is working to provide Bluetooth with a more efficient multicasting mechanism. A mechanism that is under consideration, here referred to as ACK based protocol, consists of sending the packet to a subset of slaves, known collectively as the Multicast User Group (MUG), each of which replies with an ACK, if it has already received the packet correctly, in an associated half slot. This mechanism is represented in Fig. 2. The master retransmits the multicast packet until it receives an ACK from each slave.

The protocol that we propose is based on overlapping NACKs as described is Section II, and in particular the double-NACK method described in Section III is used. GFSK is the utilized modulation, therefore some extensions to the proposed method (PSK has been taken as reference) must be done. In particular, NACKs must be composed by all ones or zeros in order to produce the same phase variation over time. Moreover, between the first and second NACK, a short sequence of bits, specific for each slave, is used to obtain the relative phase change between each pair of slaves. Each slave sends a double-NACK in a shared half slot if the actual multicast packet has not yet been correctly received. Slaves can alternatively use the second part of the slave slot to send information back to the master. This mechanism is represented in Fig. 3. The master retransmits the multicast packet only if it receives a NACK in the shared half slot.

#### B. Performance Analysis

First we compare the multi-unicast, the ACK based and the overlapped NACKs methods. For space constraints, we limit
the analysis to the throughput metric. Finally, we quantify the reliability of the overlapped NACKs proposal.

In order to calculate the throughput of the three methods, we need to define a scenario. First, for the sake of simplicity, we consider that the Bit Error Rate (BER) associated to a multicast packet is equal for all slaves (the extension of the analysis to the multi-BER case requires a more cumbersome notation without adding any relevant concept). To quantify the probabilities of sending an ACK or NACK, we consider that a slave asks for a packet retransmission only when it detects an uncorrectable error. Since the payload is much longer than the header for any packet type and, differently from the header, is not protected by forward error correction (FEC) codes, we only consider errors that occur in the payload. Therefore the probability that a retransmission is requested is equal to $P_c = 1 - (1 - BER)^{N_{\text{bit}}}$, where $N_{\text{bit}}$ is the number of bits in the payload of the considered packet type. We consider that ACKs and NACKs are always correctly received by the master. This approximation is justified for the overlapped NACKs proposal when its reliability is quantified, in fact we will see that the probability of losing a packet is very close to 0. Finally, we consider a heavy traffic scenario, where the master always has multicast packets to transmit and we assume infinite retransmission limit. Under these assumptions, the dynamics of the system can be captured through a Markov model, from which the protocol performance can be easily evaluated. Fig. 4 represents the Markov chain used to model the overlapped NACKs method for a MUG of $N$ slaves. States represent the number of slaves that have not yet received the multicast data packet, and each transition corresponds to a (re)transmission of the packet.

Fig. 5 shows the throughput of the considered methods for a MUG of 6 slaves for different packet types. The BER has been assumed equal to $10^{-4}$ which is the BER value at the receiver sensitivity level. The overlapped NACKs method has the potential to substantially increase the throughput over the multi-unicast and the ACK based protocols. In particular, with the packet type $3-DH3$, the overlapped NACK protocol is able, even in this high noise situation, to reach a throughput higher than the reference value of 500 kbps.

Fig. 6 shows the throughput of the considered methods for $BER = 10^{-4}$ and packet type $3-DH3$, varying the number of slaves from 1 to 30. The number of slaves in the piconet is actually limited to eight devices by the specifications, however results have been considered up to 30 slaves both because specifications may evolve in time and because results obtained with this case study can be utilized in other cases where the number of slaves in the MUG does not have the same limitation. Fig. 6 summarizes the advantage of a NACK based protocol compared to an ACK based proposal. In both cases, the performance degrades as the number of slaves increases, because of the increase in the number of retransmissions. However, in the ACK based protocol there is another effect that causes a further performance degradation, namely, the increase in the number of slaves yields an increase in the time interval used by slaves to reply to a multicast packet.

Fig. 7 shows the throughput of the considered methods for a MUG of 6 slaves with packet type $3-DH3$ varying the BER from 0 to $10^{-4}$. Note that the overlapped NACKs protocol is always able to assure a throughput of 500 kbps, while the throughput of the ACK based protocol is larger than 500 kbps.
only for $BER < 0.7 \cdot 10^{-4}$.

As to the reliability of the overlapped NACKs method, a C written proprietary software, able to simulate the physical and link layers of a Bluetooth point to point communication, has been modified in order to manage the overlapping of more packets coming from different transmitters. With this program, NACK missed detection probabilities have been evaluated, varying the number of slaves that are transmitting double-NACKs. Time misalignment and frequency offset between signals sent by different slaves have been considered uniformly distributed within intervals as specified in the specifications, i.e., $\pm 10 \, \mu s$ and $\pm 75 \, Hz$ respectively. Initial phases have been considered uniformly distributed between 0 and $2\pi$ while received powers have been considered uniformly distributed between $-60 \, dBm$ and $-90 \, dBm$.

Using these missed detection probabilities, the Markov chain of Fig. 4 has been modified, adding the absorbing states $L_i$, $i = 1, \ldots, N$, representing the event that $i$ slaves have not yet received the multicast data packet (and therefore each send a NACK), but the master does not recognize the corresponding overlapping of $i$ double-NACKs. In this case, the master erroneously concludes that all slaves have received the data packet, so that some nodes never receive it. Packet error rates in the average case (i.e., each slave has equal probability of losing a data packet), $PER$, and in the worst case (i.e., every time a missed detection occurs, the considered slave loses the data packet), $B$, have been analytically calculated analyzing the Markov chain. Results have been summarized in Table I for a MUG of six slaves, for different packet types and considering two values of $BER$. Even in the worst case and with $BER = 10^{-4}$, considering the packet type $3 - DH3$, that in this situation allows the highest throughput, only 1 data packet every $1/B = 625000$ is lost on average.

V. Conclusions

In this paper, we have proposed and analyzed the overlapped NACKs scheme, a simple retransmission mechanism intended for single channel multi-access wireless networks. It is suitable for both infrastructure based networks (e.g., WiMAX, LTE) as well as ad hoc networks (e.g., Bluetooth, ZigBee). As a proof of concept, we compared the performance achieved by overlapped NACKs against those obtained by an ACK based protocol recently proposed by SIG, when using Bluetooth. The study has revealed that overlapped NACKs can yield appreciable throughput improvements while keeping a high level of reliability.

We finally remark that overlapped NACKs can work in parallel with other performance-enhancement mechanisms (e.g., hybrid ARQ techniques), which may further improve the performance of the system.

VI. Acknowledgements

This work was partially supported by the European Commission within the Seventh Framework Program under the M EDIEVAL project (Grant Agreement no. FP7-258053). The authors would like to thank Dr. Pietro Capretta of ST-Ericsson, for his contributions to the research presented in this paper.

References


Table I

<table>
<thead>
<tr>
<th>$BER = 10^{-5}$</th>
<th>$BER = 10^{-4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PER$</td>
<td>$B$</td>
</tr>
<tr>
<td>$DH1$</td>
<td>$6.9 \cdot 10^{-14}$</td>
</tr>
<tr>
<td>$DH3$</td>
<td>$3.0 \cdot 10^{-9}$</td>
</tr>
<tr>
<td>$DH5$</td>
<td>$9.7 \cdot 10^{-8}$</td>
</tr>
<tr>
<td>$2 - DH1$</td>
<td>$2.7 \cdot 10^{-10}$</td>
</tr>
<tr>
<td>$2 - DH3$</td>
<td>$1.1 \cdot 10^{-8}$</td>
</tr>
<tr>
<td>$2 - DH5$</td>
<td>$3.4 \cdot 10^{-8}$</td>
</tr>
<tr>
<td>$3 - DH1$</td>
<td>$6.4 \cdot 10^{-9}$</td>
</tr>
<tr>
<td>$3 - DH3$</td>
<td>$2.4 \cdot 10^{-8}$</td>
</tr>
<tr>
<td>$3 - DH5$</td>
<td>$6.9 \cdot 10^{-8}$</td>
</tr>
</tbody>
</table>