QOS-AWARE, ADAPTIVE THROUGHPUT-ENHANCEMENT FOR CSMA-BASED MOBILE AD HOC NETWORKS

Rainer Storn (ROHDE & SCHWARZ GmbH & Co. KG
Radiocommunications Systems Division
81671 Muenchen, Germany, Email: rainer.storn@rohde-schwarz.com)

ABSTRACT

Mobile ad-hoc networks (MANETs) are considerably appealing especially for military and desaster recovery communication applications since they offer two distinct advantages: easy setup due to the absence of an infrastructure, and robustness because MANETs have no single point of failure. As MANETs generally allow a flexible number of participating nodes the preferred medium access protocol is CSMA/CA as in IEEE802.11 [1]. CSMA/CA not only supports the required node flexibility but also a mixture of traffic types. MANETs are also able to cover large distances by means of multi-hopping. Data transport via multiple hops, however, suffers from the disadvantage of throughput decrease. The main reason for this is that the radios transmit over a shared medium the access to which must be controlled to avoid transmission collisions. With increasing number of nodes and increasing traffic load the probability for transmission collisions increases which results in a decreased medium utilization due to waiting times imposed by the MAC protocol and hence reduced throughput. The throughput also decreases with decreasing data packet size. Small data packets are prevalent in voice and network control traffic with voice still constituting one of the most important applications in wireless communications, and hence the efficient handling of small packets is imperative. In this contribution several measures are proposed which have the potential to yield a significant throughput increase, especially for small data packets and under high network load while delay is affected as little as possible. The improved transmission scheme is exemplified by means of a CSMA/CA MAC protocol.

1. INTRODUCTION

One of the most important design goals for data networks is the maximization of data throughput. This is especially true for wireless networks since the frequency spectrum is a scarce resource. In mobile ad-hoc networks, a special variant of wireless networks, the requirement for throughput enhancement is particularly important because data often reach their destination via multiple hops. As wireless modems usually cannot transmit and receive simultaneously and have limited access to the tranmission medium, data throughput is further reduced, especially in MANETs where multiple hops are used.

2. THE INFLUENCE OF PRIORITIES

When the resource availability for data traffic is low it is often important to employ QoS-mechanisms [2] in order to prioritize important information over less important one. In tactical wireless networks, for example, voice is usually considered high priority while email usually has low priority. The implementation is usually done via priority queues where higher priority corresponds to a higher probability of being scheduled for transmission. High priority data packets are then inserted in a high priority queue while lower priority data are inserted into lower priority queues. As an example Figure 1 shows an example with 4 priority queues and a scheduler that causes data packet transmission to be proportional to priority where (Prio1 > Prio2 > Prio3 > Prio4).

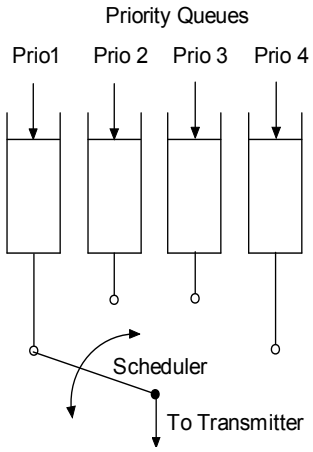

Figure 1: Example for 4 priority queues which are assigned different priorities causing high prio data packets to be transmitted preferably**.**

3. THE INFLUENCE OF OVERHEAD

In order to transmit payload data over a data network control information, or "protocol overhead" needs to be transmitted as well in order to enable the data to reach the correct destination and to check data integrity. The end user, however, has no usage for control information and hence is keen to minimize it for the sake of having more bandwidth for payload information. A popular network protocol which is used ever more frequently in the wireless domain is the Internet Protocol Version 6 (IPv6) [3], a likewise popular transport protocol is UDP. Figure 2 shows the overhead for the UDP/IPv6 protocol which amounts to 48 bytes, if no header compression is used.

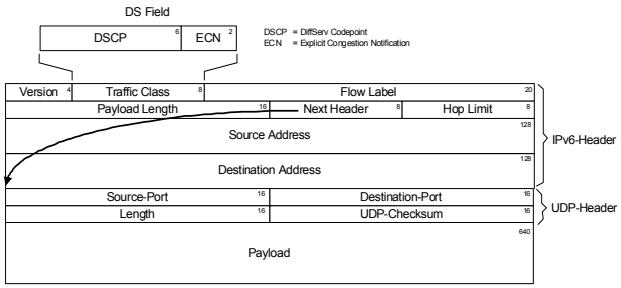

Figure 2: The overhead caused by the UDP- and IPv6-header amounts to 3*128 Bit = 48 Byte.

The overhead is increased further if the data needs to be protected, i.e. encrypted, which often is the case in the military application domain. In IP-based networks the tunnel mode of IPsec is a popular protection mechanism. The overhead for UDP/IPv6 plus IPSec header adds up to roughly 100 byte which is shown in Figure 3.

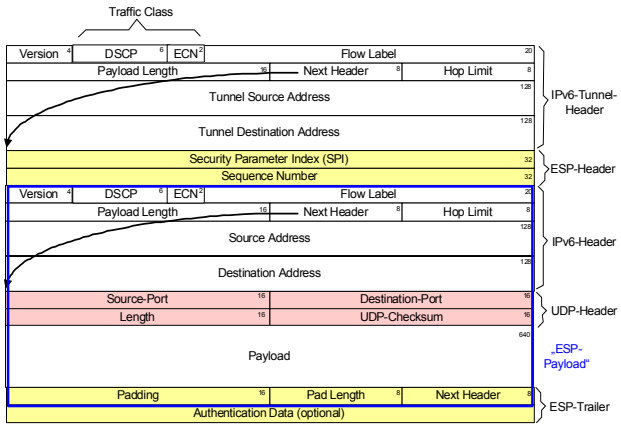

Figure 3: Ipsec-protected UDP/IPv6-data in tunnel mode. The overhead amounts to at least 100 Byte.

4. INFLUENCE OF THE MAC PROTOCOL

Before the scheduler is allowed to pick a packet from a queue and transmit it over the wireless medium the MAC functionality (MAC = Medium Access Control) needs to get access to the channel. In the case of MANETs CSMA (CSMA = Carrier Sense Multiple Access) is often used as a medium access protocol. The loss of data throughput caused by waiting for medium access can be expressed statistically by means of an efficiency factor. As an example, the MAC-protocol according to IEEE 802.11a employing CSMA/CA for MAC control and its associated efficiency factor are presented in Table 1. It is obvious that the efficiency decreases inversely to the size of the payload data packets. This is intuitive since the ratio of the data-transmit time over data-transmit-time plus idle-time decreases for smaller data packets as the idle time is independent of the packet size to a large extent.

Table 1: MAC-Efficiency for acknowledged transmission over packet size and modulation method for IEEE 802.11a.

packet size in byte	QPSK	16-QAM	64-QAM
32	15%	8%	8%
64	25%	15%	11%
96	32%	20%	15%
128	39%	25%	18%
192	48%	32%	25%
256	55%	39%	30%
512	71%	55%	45%
1024	83%	71%	62%
1500	88%	78%	70%

.

5. DATA RATE DEFINITIONS

Before focusing on the mechanisms for data throughput enhancement several important data rates need to be defined. Especially the ratio of the so called payload rate and the net data rate is of interest. The net data rate is the rate of uncoded raw bits being sent over the wireless channel. An overview of the most important data rate definitions is summarized in Figure 4. The precise definitions are given below along with a numerical example for convenience:

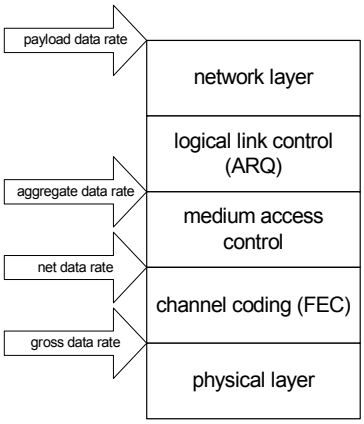

Figure 4: Definition of data rates with respect to the OSI model of communications.

gross data rate

The gross data rate for a multitone waveform such as IEEE 802.11a is determined by

$$R_b = \frac{N \cdot b_N}{T_S}$$

where N denotes the number of data carriers, b_N the number of bits per carrier, and T_s the symbol time. For the example N=48, $b_{N,max} = 6$ (64-QAM), and $T_s = 4$ µs we obtain

$$R_{b,\max} = \frac{N \cdot b_{N,\max}}{T_S} = \frac{48 \cdot 6bit}{4\mu s} = 72\frac{Mbit}{s}$$

net data rate

The net data rate R_n equals the gross data rate R_b times the code rate R_c. Hence it ensues for our numerical example in case that $R_c = 0.75$:

$$R_{n,\max} = R_{b,\max} \cdot R_c = 72\frac{Mbit}{s} \cdot \frac{3}{4} = 54\frac{Mbit}{s}$$

aggregate data rate

The aggregate data rate R_a comprises the payload rate plus the overhead rate and can be computed by the net data rate R_n times the statistical MAC efficiency K_{MAC}. Hence the maximum aggregate data rate is given by

$$R_{a,\max} = R_{n,\max} \cdot K_{MAC}$$

K_{MAC} depends on several factors, most notably the number of nodes involved in the communication. For the case of two nodes (or „one hop“), the highest possible modulation (64-QAM) acknowledged MAC transmission and a packet size of 1024 Byte K_{MAC} amounts to 62% as can be seen from Table 1.

For the example just mentioned we arrive at

$$R_{a,\max} = R_{n,\max} \cdot K_{MAC} = 54\frac{Mbit}{s} \cdot 0.62 \approx 33.48\frac{Mbit}{s}$$

payload data rate

Since the MAC protocol used in the IEEE 802.11a-example already uses acknowledgements a separate LLC is not employed. Hence the payload data rate R_p is defined by

$$R_p = R_a \cdot \frac{M_{payload}}{M_{payload} + M_{overhead}}$$

where $M_{payload}$ and $M_{overhead}$ define the number of bytes for payload and overhead information respectively. The overhead bytes are the ones being present at the MAC layer, i. e. excluding the MAC overhead. For the numerical example used so far we get

$$R_{p,\max} = R_{a,\max} \cdot \frac{M_{payload}}{M_{payload} + M_{overhead}} = 33.48\frac{Mbit}{s} \cdot \frac{1024}{1024+48} \approx 31.98\frac{Mbit}{s}$$

where $M_{overhead}$ is approximated by the UDP/IP-overhead only.

Table 2: Aggregate data rate and payload data rate over payload packet size assuming an overhead of 48 bytes, a net data rate of 56kb/s and 64-QAM modulation. The MAC efficiency is taken from Table 1, i.e. a one-hop connection is assumed.

Payload in Bytes	Payload + Overhead in Bytes	maximum aggregate data rate in Mb/s	maximum payload data rate in Mb/s
32	80	4.2	1.7
64	112	6.1	3.5
96	144	7.9	5.2
128	176	9.5	6.9
160	208	11.0	8.4
192	240	13.7	11.0
256	304	16.1	13.6
512	560	24.1	22.1
1024	1072	33.4	31.9

Table 3 shows similar data assuming two-hop communication and IPSec-encrypted data.

Table 3: In contrast to Table 2 a 2-hop-communication with encryption is assumed. Note that an overhead of 100 bytes due to encryption (see Figure 3) has to be taken into account.

Payload in Bytes	Payload + Overhead in Bytes	maximum aggregate data rate in Mb/s	maximum payload data rate in Mb/s
32	132	2.1	0.5
64	164	3.1	1.2
96	196	3.9	1.9
128	228	4.7	2.7
160	260	5.5	3.4
192	292	6.9	4.5
256	356	8.1	5.8
512	612	12.1	10.1
1024	1124	16.7	15.2

The above example shows that due to the various influences such as coding rate, MAC efficiency and overhead the payload data rate is merely 44% of the gross data rate for a payload packet size of 1024 bytes. For the often used case of voice packets the difference between the payload data rate and the gross data rate is even more dramatic due to their small size.
Table 2 shows the aggregate data rate and the payload data rate for varying payload packet sizes for a one-hop unencrypted communication.

It is evident that especially for small payload packet sizes as encountered in voice applications or for network control data the payload data rate is very small even though the net data rate has a relatively high value. The main reasons for the data rate loss and hence the throughput loss are:

1. the overhead caused by large headers of IPv6, IPsec and UDP
2. the transmission via multiple nodes
3. the low MAC efficiency (see Table 1).

5. APPROACHES FOR THROUGHPUT INCREASE

For the three issues listed above at least the following three approaches have to be implemented to mitigate the problems:

1. header compression for IP [4] and IPSec resp. HAIPE [5] to reduce overhead
2. reach enhancement by boosting the transmission power and using power efficient waveforms in order to reduce the required number of hops
3. improvement of MAC efficiency to increase the payload data rate via concatenation such as in IEEE 802.11n [6] and piggybacking [7]

Mitigation strategies 1 and 2 are traffic independent while strategy 3 is traffic dependent. In this paper we will concentrate on strategy 3 and discuss further refinements and implementation aspects. Since the MAC efficiency increases with the size of the payload packet the general approach is to increase the efficiency by sending as much data as possible without exceeding the MTU-size once channel access has been granted (cmp. Table 1) [6]. However, in the case of voice data and network control messages the delay is an important issue for reasons of service quality. Hence an adaptive aggregation scheme is imperative which allows for extra data arrival time only when necessary for the overall performance. The scheme should aggregate and concatenate data only if service quality degradation is impending, i.e. with rising number of CSMA-collisions. In fact a rising number of CSMA-collisions automatically results in a rising queue fill-level [8]. In the following an adaptive QoS-aware concatenation scheme will be presented which utilizes the queue fill-level and hence allows for self-adjustment of the concatenated data packet size.

5.1 CONCATENATION OF DATA PACKETS USING SUBQUEUES

An efficient way of concatenation is to aggregate all packets belonging to a certain "next hop" and to a specific priority queue. To this end a priority queue can be subdivided into next-hop-subqueues which are dynamically generated and removed depending on the network topology. In order to maintain fairness the packet arrivals for the priority queues have to be logged which is done advantageously in a hop-token-queue. Figure 5 shows an example of three next-hop subqueues for the hops a, b, and c and the hop-token-queue which indicates that the first packet has arrived for hop b and the second packet for hop c. The third and fourth packet in this example are again for hop b. Note that any hop token, in this example a, b, or c, can only occur once in the hop-token-queue. When the first packet for a certain hop arrives, the token of this queue enters the hop-token-queue.

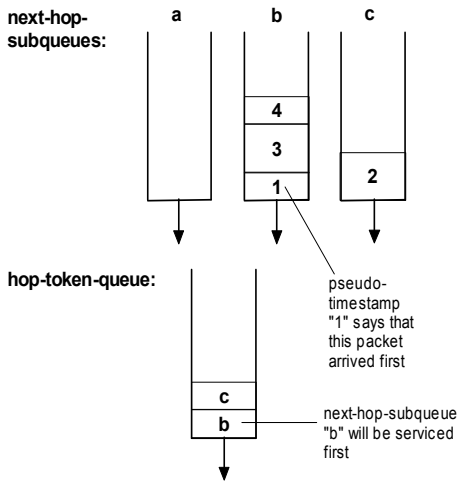

Figure 5: Detail of a priority queue makeup. It consists of next-hop-subqueues and a hop-token-queue. The pseudo-timestamp indicates the order of packet arrivals.

If a priority queue is scheduled for transmission, the hop defined by the oldest hop token, in the example of Figure 5 hop-token b, is serviced, and the packets of the corresponding next-hop-subqueue are concatenated up to the maximum transfer unit (MTU) size and transmitted. Token b is then removed from the hop-token-queue. If there are still data left in next-hop-subqueue b a new token b is reinserted into the hop-token-queue.

5.2 COMBINATION WITH PIGGYBACKING

Figure 6 shows operations needed for the packet arrival and transmit process in more detail. In addition the process for piggybacking is also depicted.

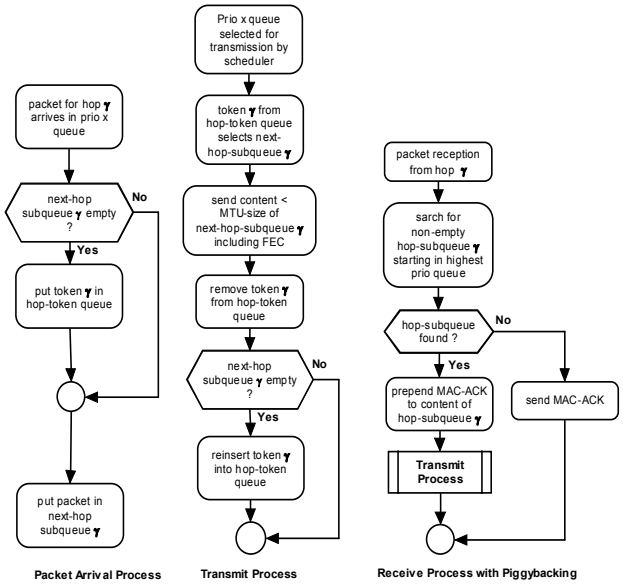

Figure 6 Processes for dealing with packet arrival, transmission and piggybacking.

Piggybacking utilizes the fact that a MAC transmission from node m to node n is generally acknowledged by an ACK message. If node n receives a message from m and notices that, in addition to the mandatory ACK, it has data for m to send these data are piggybacked onto the ACK. This way the transmission of the piggybacked data can be done without a separate contention phase leading to an increase in data throughput [6].
In the proposed transmission scheme piggybacking is combined with adaptive concatenation. Once an ACK has to be sent the next-hop-subqueues are searched, starting from the highest to the lowest priority queue, whether data can be piggybacked onto the ACK. The first hop subqueue found is selected for piggybacking (see also Figure 6). The piggybacking process is independent of the regular transmission scheduling and hence circumvents the QoS priorities. However, the potential increase in throughput by more efficient utilization of the transmission medium certainly outweighs the perturbations of QoS fairness.

5.3 MODULATION-AWARE HOP-TOKEN-QUEUE

For cognitive waveforms that are able to adapt to the current channel conditions a further efficiency increase can be achieved if the hop-tokens take modulation information into account and combine data of all next-hop-queues that use the same modulation. This way the broadcast nature of wireless transmission is exploited (see Figure 7). It is evident that modulation types may change rapidly so before each transmission this information needs to be updated in oder to decide which additional next-hop-subqueue (in this example "a") may be serviced when the primary next-hop-subqueue (in this example "b") takes turns.

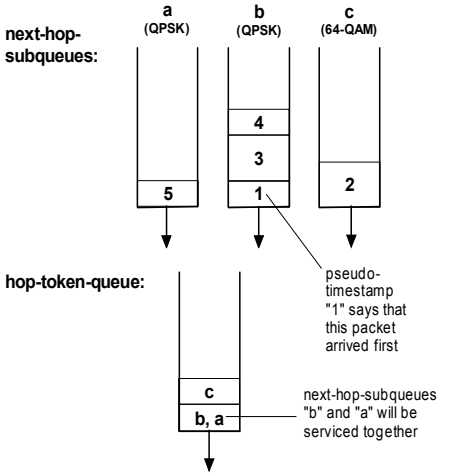

Figure 7: A modulation aware hop-token-queue allows to concatenate packets of all next-hop-subqueues that use the same modulation.

Combining all next-hop data packets in disregard of the modulation type does not seem advantageous as in this case the least bandwidth efficient modulation scheme needs to be chosen.

5.4 REDUCTION OF OVERHEAD INFORMATION

As mentioned before overhead reduction further increases transmission efficiency, so header compression or header mapping should be used [4], [5]. With concatenation, though, an additional method of overhead-reduction is to send overhead information only once for all data packets of the same type. Current concatenation approaches such as in [6] don't apply this method because the multitude of potential applications makes such an aggregation very difficult and time-variant. For military radios, however, there are a number of applications, like low bitrate voice, which are well-defined as well al frequently used. When IP and QoS are used these applications can often be identified by the traffic class field shown in Figure 2 and Figure 3. For such uniquely identifieable data, overhead redundancy can be removed. All other data must be treated conventionally. Figure 8 shows an example of overhead aggregation where a next-hop-subqueue is further divided into data-type-subqueues. If several data packets of the same type are in the corresponding queue, overhead information can be saved by sending the common overhead portion only once.

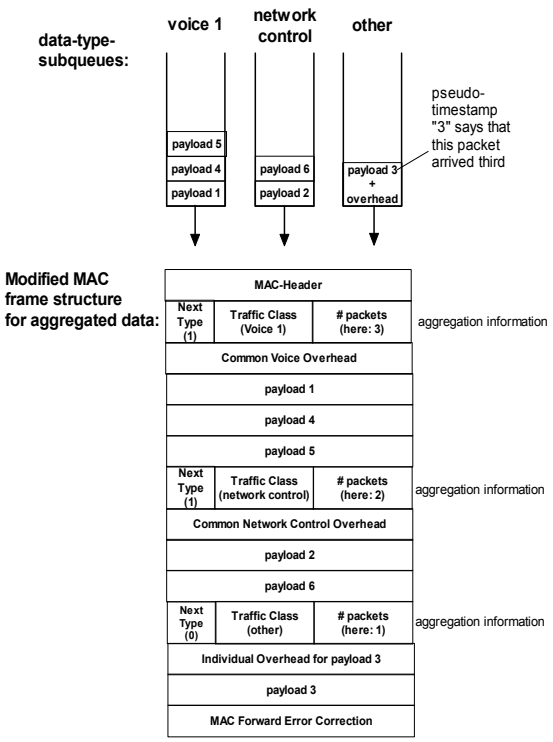

Figure 8: Proposal for dividing a next-hop-subqueue into data-type-subqueues. The common overhead portions of data of the same type need only be sent once.

Some aggregation specific overhead information must be added, however, in order to define the type and number of packets which are aggregated. Also a "next type" field is added which is set to "1" if a further packet type is aggregated in the MAC frame. Setting the "next type" field to "0" means that the following aggregation is the last one in the MAC frame

6. RESULTS AND FUTURE WORK

The proposed measures for throughput increase in MANETs while affecting delay as little as possible are multi-faceted and the achievable improvements depend heavily on the traffic mixture and load, the number of nodes and the time-dependent topology and quality of the network.

It has to be emphasized that the achievable throughput increase is adaptive and is a means towards maintaining service quality. That is for lightly loaded networks no concatenation takes place, and hence no concatenation-related throughput increase occurs. For heavily loaded networks, however, concatenation may increase strongly due to a high collision probability, which in turn increases queue fill level and so may increase the throughput by large factors, compared to the non-concatenated high collision case.

Consider the case of a voice application that generates payload packets of 32 bytes. From Table 2 and Table 3 it can be seen that the data throughput is approximately doubled if two packets of 32 bytes payload are sent instead of a single packet. This simple deduction from Table 2 and Table 3 holds if the overall overhead remains near to constant. This can be achieved by removing overhead redundancy according to chapter 5.4.

If four packets are concatenated the throughput increases roughly by a factor of four. Piggybacking may add another 50% [7] if the traffic pattern allows for it so the total throughput increase would already be sixfold. The same is true for modulation awareness the benefits of which become larger with the size of the network and with increasing traffic load, since the probability of being able to combine hops increases with the number of one-hop neighbours and traffic intensity.

To find a meaningful simulation setup will be the topic for future research that will lead to further refinements and enhancements and a more accurate quantification of the benefits of the measures proposed..

7. REFERENCES

[1] *Supplement to IEEE Standard for Information technology—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, 2007.

[2] Differentiated Services, *RFC 2474* and *RFC 2475*, Network Working Group of IETF, http://tools.ietf.org/

[3] David Morton, *Understanding Ipv6*, PC Network Advisor, Issue 83 (May 1997), pp. 17-22

[4] C. S. Tye and D. G. Fairhurst, *A review of ip packet compression techniques*, in PGNet, (Liverpool, UK), June 2003.

[5] William T. Scott, James Kohler, Greg Osborn, *Mitigating the Cost of Information Assurance in the SDR Environment*, Software Defined Radio Technical Conference and Product Exposition, Nov. 5-9, 2007 - Denver, Colorado

[6] D. Skordoulis, Q. Ni, U. Ali & M. Hadjinicolaou, *Analysis of Concatenation and Packing Mechanisms in IEEE 802.11n*, PGNET 2007, ISBN: 1-9025-6016-7, Liverpool, UK, June 2007

[7] Langguth, T., Bässler, A., Haas, E., Schober, H., Nicolay, T. and Storn, R., *A Novel Approach for Data Piggybacking in Mobile Ad-Hoc Networks*, SDR Technical Conference, Orlando 2006.

[8] D. Malone, P. Clifford, and D.J. Leith, *On Buffer Sizing for Voice in 802.11 WLANs*, IEEE Communications Letters, Vol. 10, Issue 10, pp. 701 – 703.