Multi(Uni)cast DCCP for Live Content Distribution with P2P Support

Leandro Melo de Sales
Federal University of Alagoas (UFAL) Computing Institute – IC
Maceió, Alagoas, Brazil 57.051–330
Email: leandro@ic.ufal.br

Rafael de Amorim Silva
Aeronautics Institute of Technology
Department of Electronics Engineering
Email: amorim@ita.br

Hyggo O. Almeida and Angelo Perkusic
Federal University of Campina Grande (UFCG)
Center of Electrical Engineering and Computer Science
Campina Grande, Paraíba, Brazil 58.429–140
Email: {hyggo,perkusic}@embedded.ufcg.edu.br

Abstract—Real time multimedia content transmission on the Internet is essential for the most current applications such as voice over IP, video conference, games and web TV. The most popular Internet transport protocols - TCP and UDP - do not suffice when one needs to transmit data from these applications. As a consequence, IETF has been working in new transport protocols that enhance the quality of these multimedia applications. Among all these protocols, DCCP (RFC 4340) is the most effective for multimedia content transmission on the Internet. However, DCCP is not effective in scenarios with many receivers nodes and one sender node. Therefore, this work proposes the Multi(Uni)cast DCCP, a DCCP variant that enables the multimedia data transmission from one to various nodes and supporting non-reliable traffic congestion control. The MU-DCCP uses either multicast or unicast flows according to the network support and data sharing among receiver nodes. The obtained results show that the usage of the MU-DCCP significantly reduces the data congestion in the network while improving the application scalability in terms of the number of receiver nodes.

I. INTRODUCTION

Recently, various applications have required multimedia flow in real time across the Internet. Some examples include voice over IP, video conference, games and web TV. These applications adopt protocols that enhance the quality of the data flow in addition to the effective use of their resources. Such protocols reduce the overhead of the network, preserve the equity among flows transmitted by different systems connected to this network, and ensure the minimum quality level of the multimedia content in progress.

The popularity of these applications has increased the data traffic in this network. The company Cisco Systems, Inc, predict that, in 2014, the traffic of video packets will be greater than P2P packets in the year of 2009 [1]. This corresponds to 39% of the traffic in 2011. The company also states that, in 2014, the traffic of VoIP, games, and video packets will reach the mark of 40 Exabytes/month, occupying almost 50% of the total data traffic of the Internet. Even considering the full potential of P2P services, such predictions indicate the level of inclination from the users about the sharing of their files. Recently, the company Paramount Pictures, Inc, stated its interest on transmitting movies using large P2P networks, such as BitTorrent [2]. The BitTorrent, Inc, also has announced its efforts to develop a P2P application that considers the multimedia flow in real time. Besides, the company Amazon, Inc, also announced [3] its interest in this kind of service. Hence the company will develop a fully-online multimedia renderer, enabling their users buy music and videos and watch them online supported by its cloud computing infrastructure.

Thus, the development of network protocols that support this means of providing multimedia services on the Internet has become a more complex endeavor, requiring the most effective usage of the available network resources. The Internet protocols should be designed to infer the state of the network while decisions are made considering the rapidly detectable changes. From the standpoint of the TCP/IP transport layer, traditional transport protocols like UDP and TCP have not been designed for this purpose, leaving a such task to the application developers to implement their own mechanisms for congestion control and data flows with no guarantee of delivery, particularly in the case of using the UDP, which is adopted in most multimedia applications. When an user tries to use TCP in these applications, there is no a suitable performance because the TCP implements both of congestion control and guaranteed delivery of packets in an unsatisfactory manner.

Some examples of the current transport protocols on the Internet that have been standardized by the IETF include the DCCP (Datagram Congestion Control Protocol) [4] [5] and the SCTP (Stream Control Transmission Protocol) [6]. Due to the scope of this research lies on the DCCP protocol, discussions about new transport protocols such as the SCTP will be omitted.

In this work, the application scenario involves the transmission of multimedia data in real time in addition to the usage of congestion control algorithms that share the transmission channel among all the nodes in the network. This scenario usually entails challenges such as: (i) to enable that multimedia flows co-inhabit with streams of elastic data applications without degrading them and (ii) to avoid excessive loss of data from the multimedia applications, since the retransmission of lost data is unfeasible.

Various applications can be applied to this scenario, mainly when the P2P service model is employed, such as:

- Application for multimedia content transmission in real time from a network node to another, or for a set of nodes. For instance, transmission of web pages, such as the livestream.tv and the streamtheworld.com able users to broadcast multimedia content from their computers toward to thousands of other computers connected to the Internet;
- IP telephony-based applications such as Skype, using mainly the group conversation mode;
- Online TV. As a example, the transmission of the World Cup football matches on the Internet;
- Games, video-conference, and online radio stations.

To deploy such kind of applications, at least two major issues emerge: (i) how can we avoid that application developers implement this type of mechanism for their own applications, which inevitably would increase both the mechanism complexity and its management life cycle and (ii) which actions we must take to achieve the goal contained in the first issue.

In this work, we discuss the usage of transport protocols of the TCP/IP stack for transmission of multimedia flows in real time. In particular, we present the drawbacks that involve the usage of DCCP in the aforementioned scenarios of network application and a proposal solution named MU-DCCP, the Multi(Uni)cast variant of the DCCP protocol for live content distribution with P2P support for congestion control and sharing of multimedia data among connected peers.
In the next section we provide some background on DCCP and multimedia transmission. In Section III, we will discuss details about the problem addressed in this work. In Section IV, we will introduce the main features of MU-DCCP. In Section V, we will present the methodology to obtain the results discussed in Section VI. Section VII presents related work and, in Section VIII, are presented the conclusions of this research.

II. BACKGROUND

In real-time multimedia flow applications, it is advisable to maintain the data flow and reproduce the content that is waiting rather than relaying lost information, despite the fact that part of the application data has been lost. This is not possible when using TCP because it implements reliable data delivery by adopting the method of retransmitting lost packets.

On the other side, the UDP protocol has been widely used in real time multimedia applications because it is a lightweight protocol that makes use only of the best IP service effort to transmit data on the Internet. Over the years, the UDP was the only option for multimedia data transmission in real time. However, various side effects have been verified on large network grids. These effects have been widely discussed in the literature [7]–[9].

A. The DCCP Protocol

With only these two options for transporting data across the Internet and for performing improvements on the services offered by the multimedia applications, the IETF approved the DCCP protocol specification for multimedia data transport on the Internet. It is a network protocol located at the transport layer of the TCP/IP stack, such as the TCP and the UDP. This is a connection-oriented protocol, but it does not guarantee delivery or the ordering of the data transmitted. However, it implements congestion control for transmission of unreliable data flows [8].

As a result, the DCCP inherits from the TCP the characteristics of being connection-oriented and of providing control over congestion. From the UDP, the DCCP inherits the features that guarantee neither delivery nor the ordering of the data transmitted. Apart from these features, the DCCP adds two new concepts: the late data choice as well as a framework for managing congestion control algorithms in a modular fashion. The late data choice enables the removal of data from a packet even after it has been sent to the transport layer, although it has not been sent by the network. On the other hand, the framework of the control algorithm for managing congestion enables to add new congestion control algorithms to the application and use them even when a DCCP connection has already been established.

To understand the improvements made by the DCCP protocol, let us consider the graph throughput × time, as shown in Figure 1. In this graph, we have illustrated both the TCP and DCCP behavior when used for the transmission of a file and a multimedia content. Observe that both TCP and DCCP share the available bandwidth, enabling that each flow may transmit data over the network.

This paper presents the DCCP protocol as an effective option to be adopted for the transmission of multimedia data, mainly due to its behavior (similar to the TCP) as well as for being a standardized protocol. The adoption of the DCCP improves the usage of network resources, as shown in aforementioned figures. However, it still presents some critical flaws which need to be corrected.

III. THE DCCP PROBLEM FOR DISTRIBUTING MULTIMEDIA DATA

The DCCP protocol obtains encouraging results because of its effective multimedia data transmission and better use of the network resources which may solve the most apparent TCP and UDP problems. But there is a limitation of the DCCP protocol when used for distributions of multimedia content, being transmitted from one network to various nodes located on different networks (1 → N).

It is unfeasible the usage of the DCCP protocol to perform this type of transmission because the DCCP is a connection-oriented protocol, and consequently, for every new user interested in receiving a multimedia flow transmitted with this protocol, a novel connection is necessary. The consequence of this DCCP limitation can be disastrous, and may result into a paradoxical protocol to accomplish its major goal: to address network congestion generated by the UDP protocol for multimedia applications in various scenarios.

When the protocol is used to transmit to a big range of clients, it becomes ineffective because the transmission rate of sender node reduces to zero. DCCP performs congestion control by using an equation that defines the connection transmission rate. The more nodes get connected to a transmitter node, the lower will be the rate of transmission of the transmitter node for each one of the receivers connected to it. For the network, this strategy seems to be fair because it avoids the network to collapse under congestion, but for all the receiver nodes this is most prejudicial. This phenomenon has been widely described in the literature as the tragedy of the commons [10]. The tragedy of the commons did occur in this case because the transmission rate of a given flow become too low as new receivers connect to the transmitter node. As a consequence, it may not have been a good throughput from the sender enough to justify the reception of a multimedia content in real time. Statistically, the rate of transmission in each DCCP flow converges to a point of equilibrium. However, this point will be from minimum capacity, not enough for the scenarios of applications considered in this work, although all receivers have been assigned with equal rights while using the channel.

This problem can be described mathematically by using, as a support, Equation 1 [11], which defines each $X_i$ transmission rate calculated by the DCCP CCID-3. In this equation, $X_i$ is the transmission rate in bytes/second, $s$ is the packet size in bytes, and $R$ is the RTT in seconds, $p$ is the occurrence rate of losses between 0 and 1. $RTO$ is the timeout value of the TCP retransmission in seconds and $b$ is equal to 1, representing the maximum number of packets supported by just one ACK. Let us consider the problem described above, defining the use of the channel for a total amount of $N$ DCCP flows by $B = \sum_{i=1}^{N} X_i$. In severe conditions of network congestion, the value for $b$ is equivalent to the channel bandwidth. When this occurs, we understand that $N$ has reached a value greater than...
the network can support, causing the value of \( p \) and \( R \) in the Equation 1 to increase, and producing as a result \( \lim_{N \to \infty} \frac{B}{N} = 0 \).

\[
X = \frac{R \times 2 \times 5 \times \frac{1}{2} + (RTO \times 3 \times 3 \times 0.5 \times x \times (1 + 32 \times x^2))}{1}
\]

IV. MU-DCCP: MULTI(UNI)CAST DCCP

Multi(UNI)cast Datagram Congestion Control Protocol (MU-DCCP) is an extension of the DCCP protocol for transmitting multimedia data flows in scenarios whose a node can transmit to \( N \) receivers. This extension enables the transmission of data packets in order to support congestion control of unreliable flows. The MU-DCCP can operate into two modes of transmission: (i) multicast, and (ii) multi-unicast. In both modes, the application selects the desired mode and a specific algorithm creates a connection that establishes data transmission between \( N \) receivers. The multicast mode is commonly used in local networks, whereas the unicast mode is used by a special node called MU-DCCP Relay within a local network that connects to the source node and redistributes the content in its local network.

A. MU-DCCP overview

When an application initializes a MU-DCCP socket corresponding to a desired connection, it sends a packet of the type DCCP-MREQUEST with the TTL field of the network layer equal to 1. This packet is sent on multicast mode with the address 239.255.255.251 and the port 1900 (this socket is called control channel of the MU-DCCP).

In the packet DCCP-MREQUEST, there are two fields: the first one specifies the transmitter IP address (32 bits) and the second one specifies the port (16bits) of this transmitter. As the packet is transmitted across the local network with TTL=1, it cannot be routed to the external network, and just the node of the local network can receive it. Denote the address 200.200.211.5 and the port 8900 as the socket of the transmitting node which the receiving node wants to be connected. In this case, the receiver sends a multicast packet for the multicast address previously specified and with both IP and port fields fulfilled with data for the socket of the transmitter node.

On the other hand, all other existing DCCP nodes at the local network should, as implementing the extension of the MU-DCCP, start a multicast socket with the IP address 239.255.255.251 and port 1900. This enables the reception of DCCP-MREQUEST packets transmitted by any network node that needs to receive some specific multimedia content. Consequently, if a local network node is receiving a multimedia DCCP data flow proceeding from address 200.200.211.5 on port 8900, it will reply to the interested node in a packet of the type DCCP-MRESPONSE upon receiving a DCCP-MREQUEST. This occurs in the unicast mode.

At replying with the DCCP-MRESPONSE, the receiving node of the original flow starts to operate as a relay node (MU-DCCP Relay), redistributing the packets that it receives from the sender or from another relay node. At the DCCP-MRESPONSE header, the node specifies which mode of transmission (multicast or unicast) that it will relay the packets across the network. As in the first time there is no node of the type MU-DCCP Relay on the local network, the node interested in receiving the flow can send a packet of the type DCCP-MREQUEST, but with TTL = 2. If the local network router is routing multicast packets towards to the IP address and the control channel port of the MU-DCCP, it is possible that a MU-DCCP Relay replies with a packet of the type DCCP-MRESPONSE in the means that has already been demonstrated.

If the node that sends the DCCP-MREQUEST fails to receive any type of packet of the type DCCP-MRESPONSE, it will have two options: either the node will send another packet of the type DCCP-MREQUEST with the TTL incremented by 1, or the node will start a unicast connection with the source node. Both ways are configurable through the configuration parameter of the application (set up via setsockopt) function, from the BSD Socket API. In practice, we recommend that if a MU-DCCP client do not receive any DCCP-MRESPONSE packet within 3 attempts (TTL=3), it initiates a connection process with the source node. Note that the MU-DCCP node that starts a unicast connection with any transmitter nodes (another MU-DCCP relay or the source node), it will elect itself to become a MU-DCCP Relay. In this case, for future DCCP-MREQUEST packet it receives, it will reply to the requester with a packet of type DCCP-MRESPONSE, as described before.

B. MU-DCCP transmission modes

The reply to a connection message sent by a MU-DCCP (using DCCP-MREQUEST packet) contains three fields: (i) the multicast field (1 bit); if enabled, the node will transmit all data on the multicast mode; (ii) the IP address field, which specifies the IP Address (32 bit) that the MU-DCCP Relay will start transmitting the multimedia content or it is already transmitting to another DCCP nodes; and (iii) the port field (16 bits) which the MU-DCCP Relay will transmit the data through a server socket or a multicast socket with the source node.

It is important to state that a MU-DCCP Relay may decide to change the mode of transmission at any time by simply sending a DCCP-MSYNC packet via multicast to the control channel of the MU-DCCP with the desired modification. On receiving a DCCP-MSYNC, the busy node with the reception process of the data will make the necessary modifications to its way of reception. The format of the DCCP-MSYNC is the same in relation to the packet format DCCP-MRESPONSE.

C. DCCP congestion control on multicast mode

The MU-DCCP supports data transmission on both unicast and multicast modes. Control congestion algorithms on the multicast mode are often more complex because they use feedback from receivers to make decisions about the configuration of the transmission rate. In this case, no CCID for the DCCP has been designed to operate on this mode. Therefore, it is relevant to work on a new CCID able to dealing with all features of non-reliable data flow transmissions on multicast mode. Within the scope of this work, we defined the CCID-5, an algorithm for congestion control of unreliable data flows in multicast transmissions using the DCCP protocol.

The main focus of this algorithm is to run it on a MU-DCCP Relay that is being transmitted on multicast mode. The CCID-5 determines a new rate of transmission based upon reports sent by receivers of the local network. Because only non-reliable data flows are being treated, the solution for CCID-5 is to make use of this type of transmission to reduce the necessity of confirmation for all packets received by receivers. As we are only considering scenarios where there is one data transmitter node for various receiver nodes, to receive confirmation of reception might flood in the transmitter control packets, a situation known as feedback implosion.

As a consequence, the CCID-5 algorithm is a congestion control algorithm that does not necessarily need to receive a receipt report from all the receiver nodes, but from a subset of receiver nodes. It reduces the amount of control data transmitted over the network. Note that the transmitting node is not only the one that generates content, but also is able to be a node of the type MU-DCCP Relay. Therefore, we developed an algorithm to enable the elections of MU-DCCP Relys and MU-DCCP Reporters nodes, responsible for
reporting to a transmitting node the data reception rate through multicast. Upon receiving reports from MU-DCCP Reporters, the MU-DCCP Relays will adjust their rate of transmission in accordance with these reports. This is done by using the Equation 1 for the calculus of transmission rate.

D. Election of MU-DCCP nodes

The MU-DCCP Relay nodes are selected into two means: (i) MU-DCCP Relay nodes start the first unicast connection with a data generator DCCP node i.e., in the original sender node; (ii) the MU-DCCP Relay nodes are capable of negotiating with another MU-DCCP Relay node its promotion for becoming a MU-DCCP Relay. The node that enables the promotion of a MU-DCCP Relay to another node must bring itself to a lower MU-DCCP common node. At the same time, when a MU-DCCP Relay passes this status to another, it can keep disconnected from the DCCP generator node by sending a DCCP-MRESET containing the address of the new MU-DCCP Relay. It is also possible that a MU-DCCP Relay node promotes others to the status of secondary MU-DCCP Relay located on its own local network. This feature is important because whether the current MU-DCCP Relay loses its connection or disconnect itself from the data generating transmitter node, any secondary MU-DCCP Relay will be able to take over the role of the primary MU-DCCP Relay. In this case, the node that controls this role will send a packet of the type DCCP-MELECT informing that it will monitor data transmission which had been previously provided by the old MU-DCCP Relay.

In relation to MU-DCCP Reporters, the election process works in a similar manner. However, any node can become a MU-DCCP Reporter. In summary, the process works as follows: When a MU-DCCP Relay receives packets of the type DCCP-MREQUEST, the node MU-DCCP Relay activates a bit size field flag in the the DCCP-MRESPONSE informing that the node has to send reports to the MU-DCCP Relay. Thus, the MU-DCCP Relay can limit the number of MU-DCCP Reporters and thus receive reports only from a subset of the network. When the transmission is multicast at local, it is possible to determine the best transmission rate. Within the context of this work, only a few experiments were conducted as an attempt to discover whether it is possible to determine the number of MU-DCCP Reporters to obtain correct transmissions rates.

E. Multimedia flow adaptation

A peculiar feature of the MU-DCCP is the ability of adapting multimedia flows in a distributed manner. Most solutions for multimedia data transmission, apart from establishing congestion control at the application level, they solve adaptations of multimedia flows at the data generating source. A solution for multimedia flow adaptation is to broadcast data for different channels, where multimedia flows are transmitted with different levels of quality. Depends upon the quality desired by the receiver node, it requests transmission to a certain channel. The problem of this approach is that the transmitter node must relay data into multiple channels, an operation that invariably increases the application complexity and the quantity of data flows being transmitted from the server.

In MU-DCCP, it is possible to run the adaptation of data flows in a minimum manner in every MU-DCCP Relay. Let us suppose that there are two adjacent networks: network 1 and network 2. Consider that there is one network in the MU-DCCP Relay of the network 1 and that between this network and the transmitter node there is a bandwidth of 100 Mbps. If the bandwidth on the network 2 is of 10 Mbps at its maximum capacity, a receiver node on the network 2 would have to ask for a multimedia flow along a different channel, considering the aforementioned solution. In the case of MU-DCCP, it is possible that a node on the network 2 can obtain the multimedia via the MU-DCCP Relay present on the network 1, which can occur whether the MU-DCCP Relay on the network 1 adapts the flow for the node found on the network 2. In this way, one can reduce the traffic in the network of the transmitter node and still allow for other nodes in networks with low bandwidth to obtain adapted multimedia flows.

V. METHODS AND EXPERIMENTS

Due to limitations regarding computational resources to execute experiments related to the problem described above, simulations are done by using NS-2. We performed simulations on 10 network scenarios, considering a binary tree topology, where each branch of the tree represents a router, and each router has 10 DCCP nodes connected to it. Each one of the 10 scenarios are repeated n times until an average confidence level of 95% be achieved. For each scenario, the level of the binary tree defining the topology of a network was increased by 1. For example, the first scenario had 10 receiving nodes and a router; in the following scenario, it had 30 nodes and three routers; in the following it had 70 nodes and 7 routers, and so on. The network topology is illustrated in the Figure 2, while the settings were defined as follows:

- Number of receiver computers per network: 10
- Local network bandwidth: 100Mbps
- Local Network Latency: 1ms
- Bandwidth of backbone: 300Mbps
- Backbone Latency: 25ms
- Queue size of routers on the backbone: 3000 packets
- Duration of simulation: 900s

![Fig. 2. Network topology used in this study. Each node represents a router connected to 10 nodes.](image)

The transmissions were performed as follows: one node, located at the root of a tree, sent the same multimedia content to all others connected to the network, simulating, in this way, a typical multimedia transmission \(1 \rightarrow N\) with a data traffic pattern equivalent to VoIP application. The simulations results are summarized in Section V.

VI. RESULTS

Firstly, we have conducted simulation using the standard DCCP implementation and in a second moment the same set of simulations were repeated, but using the MU-DCCP.

Figure 3 summarizes the simulation execution conducted by using the DCCP standard. The X axis of the figure represents the number of receiver nodes as the simulations were implemented; while the Y axis represents the mean transmission rate obtained by each DCCP connection.

Note in Figure 3 that the mean throughput for each DCCP flow bends towards zero as the number of receivers increases. In other words, the DCCP protocol does not scale when used
Fig. 3. Transmission of VoIP data using DCCP with one transmitter to many receiver nodes. The mean for throughput is bend towards 0 as the number of receiver nodes increases.

The DCCP protocol works most efficiently in clear-cut scenarios, but it has a plain scalability problem, which is critical for the applications considered in the present work. This renders the DCCP protocol useless for scenarios of multimedia content distribution, making developers uninterested in effectively using the DCCP protocol on their applications. Thus, the DCCP protocol is seen as an alternative to UDP in simplistic multimedia transmissions scenarios, but in situations of real multimedia applications scenarios such as found on the Internet, the DCCP protocol, as it is currently employed, cannot be used in more complex multimedia applications scenarios. Obviously, the Figure 3 shows that the more flows receivers node starts, the more pronounced will be the loss of data, causing a decrease in each flow throughput.

Considering this, we have conducted simulations with the same network topology, but using the MU-DCCP. The summary of the simulation is depicted in the graph shown in Figure 4. Observe that when comparing the graph of Figure 3 with the simulation result of the Figure 4, the usage of MU-DCCP greatly improved the effective rate of receipt of the data on the part of the receiver nodes.

Finally, another important aspect is the delay of the data reception that a receiver observes while receiving data from the transmitter. In this way, we have studied the behavior of the standard DCCP and of the MU-DCCP regarding this aspect. The graph illustrated in the Figure 5 summarizes the behavior of each protocol. The X axis represents the distance of a group of receivers (the level of the tree) to the transmitter located in the root of the tree; while the Y axis represents the mean of the reception delay observed by each DCCP and MU-DCCP connection for the respective level of the tree.

Note that, as the distance from a receiver node increases (higher levels of the tree) to the transmitter node located in the root of the tree, the reception delay for each node using standard DCCP implementation increases exponentially, while using MU-DCCP the increasing is linear. In fact, the reason for this behavior is because of the usage of combination between multicast and unicast transmission modes implemented in MU-DCCP: since it opens only one unicast channel between two MU-DCCP Relays located in two different networks and each MU-DCCP Relay distributes the content locally using multicast. Hence, while using DCCP we observe the sum of the delays due to the connection of each flow, using MU-DCCP this is reduced to the delay of only one flow, plus the delay for processing and forwarding packets by a MU-DCCP Relay.

In addition, MU-DCCP provides a mechanism that allows a MU-DCCP Relay to monitor the delay observed and, in case this delay is higher than a threshold specified by the application via BSD Socket API (using the setsockopt() function), it automatically initiates a unicast connection with the root transmitter. This is very important for the protocol scalability in terms of the number of receiver nodes: the former MU-DCCP Relay become able to process and forward packets to other MU-DCCP Relays located in higher levels of the network tree. In other words, it becomes able to forward packets to receiver nodes located farther than the MU-DCCP Relay that initiated a new unicast connection with the root transmitter.
VII. RELATED WORK

The works available in [12], [13] and [14] investigate the usage of DCCP for real-time multimedia transmission and quality of service in cable and wireless networks. Other works [15], [9] [16] and [8] address issues about DCCP performance and future work. The authors in [17], [18], and [19] investigate the adaptation of multimedia flow considering the DCCP protocol. The authors in [20] and [21] describe the DCCP protocol implementation and their programming libraries. The authors in [22] and [23] address some drawbacks of the DCCP standard version and propose some improvements toward to the solution of these problems. Other authors in [24], [25] and [26] provide evaluation results and address DCCP improvements for wireless sensor networks.

To the best of our knowledge, there are no works that address and investigate both the performance and DCCP scalability by considering the distribution of live multimedia content in scenarios with one source node transmitting data to several receiver nodes (1 → N).

VIII. CONCLUSION AND FINAL REMARKS

In this article, we have presented the results and discussions about a DCCP variant named MU-DCCP, a modified version of the DCCP with support to multimedia data transmission by combining multicast and unicast modes to transport data. This is the first work that discusses the usage of DCCP for content multimedia distribution, once previous works only addressed DCCP issues in terms of its performance regarding to the congestion control algorithms for unicast traffic and fairness with other protocols, such as TCP.

Some conclusions were addressed to the use of the DCCP protocol for the transmission of data in large scale application of scenarios 1 → N: (i) the protocol DCCP is not satisfactory in such scenarios. For example, in applications of multimedia content distribution stemming from a transmitter node; (ii) solutions are then required to solve the problem affecting the DCCP protocol, and the MU-DCCP was the answer to it. Following the definition and implementations of MU-DCCP, several simulations were performed with the MU-DCCP, with the main results and further discussions being presented in this article. In the context of MU-DCCP, it has also been developed the CCID-5, a congestion control algorithm for the transmission of multimedia data with multicast support. According to the results obtained, a significant improvement in the use of DCCP in real-time has been observed.

As a future work, we intend to write an Internet Draft for later submission, such as the RFC promoted by the IETF. As the results have been most promising, it is expected that the MU-DCCP be adopted by major multimedia applications in the near future. The MU-DCCP proposal is important because it abstracts all the complexity of data transmission with a congestion control mechanism and support to multicast mode, a feature that does not exist in the original DCCP protocol.

We also intend to develop the MU-DCCP to improve the algorithm of the election of MU-DCCP Relays and MU-DCCP Reporters. This improvement is linked to the execution of more simulations on different scenarios from those studied and presented in this work. Moreover, we also will run other simulations on MU-DCCP for the transmission of data with different traffic patterns such as video clips and games, and to perform further studies on the delay occasioned by receiver nodes as they get away from the transmitter node in the network.

REFERENCES

[1] N. Leavitt, “Network-Usage Changes Push Internet Traffic to the Edge,” Computer, vol. 43, no. 10, pp. 15–16, 2010. [17] [18], and [19] Investigate the adaptation of multimedia flow considering the DCCP protocol. The authors in [20] and [21] describe the DCCP protocol implementation and their programming libraries. The authors in [22] and [23] address some drawbacks of the DCCP standard version and propose some improvements toward to the solution of these problems. Other authors in [24], [25] and [26] provide evaluation results and address DCCP improvements for wireless sensor networks.


