Exploiting Cognitive Radios in Tactical Point to Multipoint communications

Jawad Seddar, Hicham Khalifé, Vania Conan and Jérémie Leguay
Thales Communications & Security - France
Email:{name.surname}@thalesgroup.com

***Abstract*—Spanning several frequencies, cognitive radios support dynamic management of the rate vs. range requirements of new generation tactical radio networks. We exploit this agility to offer point to multipoint transport implementing group communication for data services in the network. The key feature of our proposed transport protocol (PMT) resides in its ability to convey common traffic to multiple users, while at the same time carrying information to each user as quickly as possible. This is achieved by performing high level clustering of receivers in homogeneous groups, each group being served at a suitable throughput. We present an implementation of the PMT protocol in software over a GNU radio based hardware and demonstrate the performance enhancements this brings in the case of two groups of receivers, with nodes operating dynamically on different frequencies. It is therefore fully suited for novel data centric services mandated by the next generation tactical networks.**

I. Introduction

The next generation of tactical networks will support new forms of operational engagement, such as Network Centric Warfare [1]. Massive transformation programs are following this path in the US [2] and in Europe [3]. These new programs are mainly driven by the increasing needs for high bitrate data communications to support new services such as Blue Force Tracking (BFT), multimedia content delivery and remote control of sensors. However, as commonly known the major challenIn this paper we thoroughly study over a software defined radio platform the case where receivers fall naturally into two groups. This may correspond typically to a tactical network where receivers can be reached with two different waveforms.ge and difference to overcome in tactical communications reside in providing these services in a point to multipoint mode where receivers are distributed following a military hierarchy (battalion, company, and platoons) or for mission centric services cutting across the hierarchy.

In practice, such innovations are becoming possible with the design of innovative waveforms [4] and the emergence of intelligent and agile radio devices [5]. These programmable radios able of interference sensing, environment learning, and dynamic spectrum access lead to the so-called cognitive radio technology [6] that promises to improve spatial reuse and observed throughputs. In the particular domain of tactical communications, cognitive radios need to dynamically decide between, *i)*choosing traditional tactical VHF frequencies, that are able to perform long range communication, at low or moderate bitrates or alternatively, *ii)*selecting higher frequencies (upper VHF or UHF) that support higher bitrates, at the expense of coverage range. Nevertheless, in a point-to-multipoint context, users can be spread over different frequencies, channels, or locations. Thus, the links “connecting” each destination to the source, might have inherently different characteristics (e.g. bandwidth, center frequency thus propagation properties) or be interfered by different (primary or secondary) users. As a result, users in the same group may experience very heterogeneous performance in terms of latency, physical transmission rate, MAC layer retransmissions, etc.

This work is supported by the French National Research Agency (ANR) under grant reference LICoRNe ANR-10-VERS-005

To support point-to-multipoint distribution of the same data across heterogeneous receivers, the source can adapt its flow to the slowest receivers, like in the NORM [7] protocol or in RTMP [8]. This however translates in pulling down the reception rate of all nodes of the group i.e following the rate imposed by the lowest frequency; while this might be appropriate in the Internet where receivers have close behaviors, it is clearly not the case in a multi-channel wireless context. Alternatively the source could follow the fastest receivers, but it is taking the risk of “losing” the slowest, resulting in too many packets being dropped on the saturated slow channels.

To solve this dilemma, we have proposed PMT (Point-to-Multipoint Transport)in [9], an acknowledgement based transport protocol which dynamically differentiates among receivers and separates them according to their reception capabilities. PMT creates dynamic groups of receivers managed by the source to improve delivery time for the nodes that can receive data early, and thus the overall throughput.

In this paper, we prove the correctness of our solution and its feasibility, by implementing our protocol over a cognitive radio platform deployed in our premises. We focus on the case where receivers fall naturally into two groups. This may correspond typically to a tactical network where receivers can be reached with two different waveforms. Our platform, based on GNU radio devices, reproduces tactical point to multipoint environments. In practice, our experiments highlight the fact that PMT can cope dynamically with link property variations thus adapting automatically the group members as well as the transmission rate of every group.

The remainder of the paper is structured as follows. Section II. details the protocol and its mechanisms. Section III. describes our cognitive radio platform. We define and analyze our experimentation in IV. Finally, related work is given in V then conclusion and future work in Section VI.

II. Transport protocol

A. Preliminaries

In a multidestination configuration, the throughput of a group composed of N members can be expressed as the sum of the throughput of all the members of the group.

$$\Phi_{group} = \sum_{i=0}^{N} \Phi_i$$

where Φ_i is the throughput observed by the member i of the group.

In order to prevent slow receivers from penalizing those benefiting from favorable network conditions, we seek to create dynamically separate groups each served at a particular throughput. We base the group formation algorithm on the round trip time (RTT) observed by each node. With these observations the source node is able to differentiate between slow and fast nodes. A single time threshold T is sufficient to discriminate between both groups: all nodes below the threshold T go in the fast group, all nodes above in the slow group (their RTT is larger). In the considered setting, the tradeoff is the following: one could try to maximize the number of receivers in the first group. However, this would imply increasing T and thus reducing throughput for all nodes in the first group. It is intuitive that an optimal value of T should exist, depending on the RTT probability distribution. Please refer to [9] for a detailed theoretical analysis.

B. Protocol description

Our mechanism is source driven, in other words the source node maintains, in a special database, the group affiliation for every receiver. The average RTT for every receiver (other participants within the zone) is also stored inside this database. Moreover two transmission buffers are added, each handling transmissions for a precise group. The protocol building blocks are shown in Figure 1.

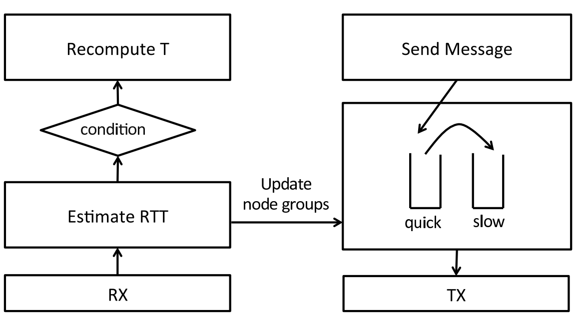


Fig. 1: Protocol building blocks

The source sends a message for the fast group every T and serves the slow receivers every T_{max}. In fact, the source transmits the message available in the fast nodes queue to the fast receivers and waits for the acknowledgements. After T seconds (i.e at the expiry of the fast nodes interval), receivers that have answered are labeled as fast; all others are labeled as slow in the specific database. Using timestamps, the smoothed RTT of fast receivers is also updated. The message is then transfered to the slow group buffer and transmitted to the slow receivers at T_{max}. Note that T_{max} here is a fixed protocol parameter, dictated by the application requirements, or the need to ensure quite slow nodes can still get enough packets. In order to better illustrate our approach, we choose to handle here the simple case of two groups and fixed (large) T_{max}, and defer the more general cases to future work.

This flow control process is repeated whenever new messages are available for transmission. More generally, the throughput of our protocol is dictated by T and T_{max} as follows:

- at T source pushes the packet to the slow group queue, pops a new packet and sends it to the fast group receivers.
- at T_{max} sources removes from the slow queue the packet sent T_{max} seconds earlier, then transmits the packet in head of queue to the slow group members.

Since the objective is to handle dynamically rate vs. range requirements enabled by cognitive radios, the strategy of the protocol is to improve delivery time for the nodes that can receive data early. The long-term throughput of the system is unchanged, as it is dictated by the second queue (the slow nodes that are served every T_{max}), since all nodes receive the same data.

C. Algorithm for dynamic group calculation

In order to select the appropriate value of T that separates the fast from slow nodes, we propose a greedy algorithm which maximizes the average throughput per node (which is the same as maximizing total network throughput for a fixed number of receivers).

Algorithm 1 Estimate optimal value of T

Input: N //total number of receivers
$\tau[N]$ //table containing smoothed RTT of every receiver
T_{max}
$max\ = 0$, $index$, $result$, j //intermediate variables
Output: T

1: sort($\tau[N]$)
2: **while** $j\ <\ N$ **do**
3: $result \leftarrow (j.\frac{1}{\tau[j]}) + (N-j).\frac{1}{T_{max}}$
4: **if** $result > max$ **then**
5: $max \leftarrow result$
6: $index \leftarrow j$
7: **end if**
8: $j = j+1$
9: **end while**
10: $T \leftarrow \tau[index] + \epsilon$
11: **return** T

The basic idea of the algorithm is to determine the value of T by computing the average throughput based on the receivers' RTTs. First, we start by sorting received RTTs in increasing order (line 1). Then, by sequentially selecting the

RTT of receiver j and computing the throughput of each group accordingly (i.e by also including all receivers having smaller RTT) we estimate the throughput as if the RTT of receiver j equals the value of T (line 3 of the algorithm). At the end of this loop the algorithm returns the RTT value that offers the highest total throughput. In practice, a slightly bigger ($+\epsilon$) value from this RTT is selected for T in order to maximize the total network throughput. In fact, this small margin allows to account for potential RTT fluctuations.

III. Cognitive Radio Platform

We describe herein the cognitive radio platform we use to evaluate the performance of the PMT protocol.

A. Software Defined Radio Devices

In order to implement the previously detailed protocol, we use the Universal Software Radio Peripherals (USRP) made by Ettus Research [10]. In our tests, we rely on USRP1 devices, which are the first generation of the USRP products commercialized by ettus.

The USRP1 is a radio device built around a FPGA. It possesses four 12 bit Analod-to-Digital Converters (ADCs) running at 64MSamples/s and four 14 bit Digital-to-Analog Converters (DACs) operating at 128MSamples/s. This enables us to have four complex channels simultaneously (4 I channels and 4 Q channels). Therefore, up to two complex inputs and two complex outputs can be simultaneously exploited.

This software defined radio is controlled through particular softwares running on a computer (described later). The communication to the computer is done through a USB 2.0 connection linking the computer directly to the FPGA through a Cypress FX2 USB controller. The USRP1 motherboard has four extension slots on which several kinds of daughter boards can be plugged. In our experiment setup, we use 2 daughter boards of 2 slots each in order to fill the all 4 available extension slots

- The RFX900 which enables us to transmit and receive around 900 MHz (GSM frequency)
- The RFX2400 which enables us to operate around 2.4 GHz (ISM band)

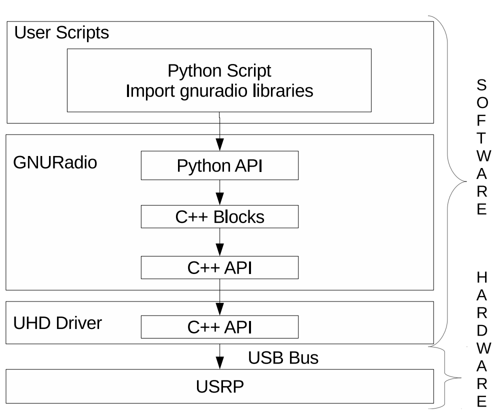


Fig. 2: USRP & GNU Radio software stack representation

B. GNU Radio

GNU Radio [11] is a free and open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used with readily-available low-cost external RF hardware to create software-defined radios, or without hardware in a simulation-like environment.

GNU Radio applications are primarily written using the Python programming language, while the supplied performance-critical signal processing path is implemented in C++ using processor floating-point extensions, when available. Thus, the developer is able to implement real-time, high-throughput radio systems in a simple-to-use, rapid-application-development environment.

GNU Radio is therefore a very useful software tool but it needs an additional layer in order to control our software defined radio devices (III-A).

This is the role of the USRP Hardware Driver (UHD) provided by Ettus Research. It is provided as a standalone driver, and is made available to the GNU Radio toolkit through the implementation of several blocks, such as an emitter (uhd.usrp_sink), a receiver (uhd.usrp_source), etc.

Therefore, the overall system architecture can be thought of as a stack with the hardware (USRP device) sitting at the bottom of it. UHD is the direct link to the hardware and GNU Radio is the link between user defined flow graphs and UHD. Although one could directly connect to the hardware through UHD and without the use of GNU Radio, they would be limited to simple operations while the GNU Radio toolkit is very furnished. The complete hierarchy is shown in Figure 2.

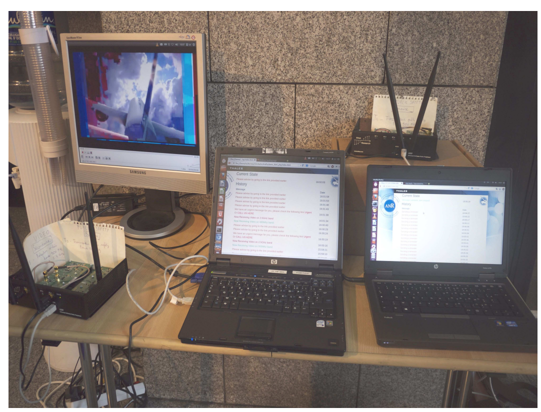
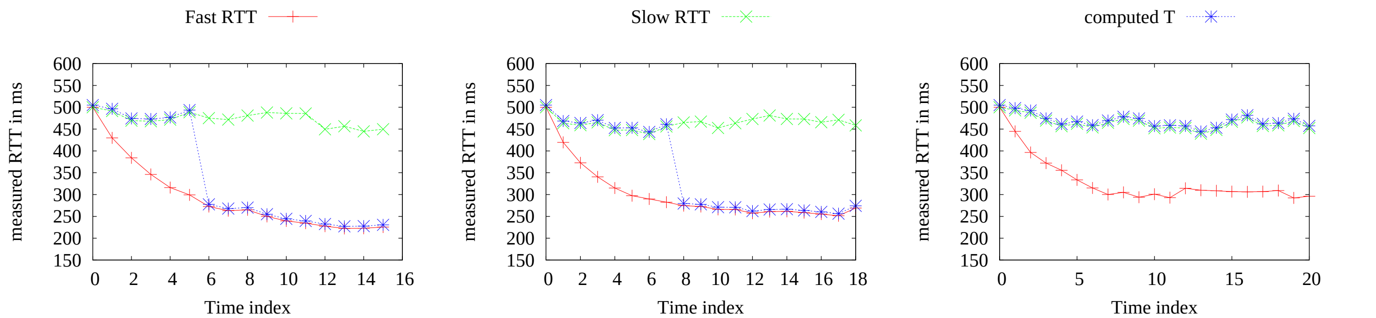


Fig. 3: Deployed GNU radio platform

IV. Experimental Results

A. Considered Setup

The experimental validation of the PMT protocol requires at least 3 nodes: 1 emitter and 2 receivers. Because one can have several emission and reception channels on a given USRP, we used one USRP as the emitter and a second USRP as the receiver, both with two channels for emission and reception. More precisely, the source transmits simultaneously



(a) 500 Kbit/s (b) 320 Kbit/s (c) 250 Kbit/s

Fig. 4: T values for different capacities.

over 2 channels whereas a single USRP1 plays the role of 2 receivers by listening and reacting separately over the 2 channels exploited by the emitter (refer to Figure 3 for an example).

- The first channel is a ”high” capacity channel with a capacity ranging from 250 Kbits/s to 500 Kbits/s and a working frequency around 2.4 GHz.
- The second channel is a ”low” capacity channel with a capacity of 125 Kbits/s and an operating frequency around 900MHz.

This one hop, multiband and multipoint setup enables us to test the protocol in different rate vs. range conditions where our mechanisms in place can either recognize two groups or one group to optimize the global throughput.

In practice, we put in place the following 3 steps scenario to test the PMT protocol:

1) The source sends 1 Megabytes file at the rhythm of 4 packets of size 1024 Bytes at a time to its 2 receivers and then expect an acknowledgement.
2) Upon each acknowledgement reception, the source measures the RTT for each destination and computes a smoothed version of it.
3) It then computes a new value for T, using our defined algorithm, based on the smoothed RTT for each destination.

Note that we consider a static T_{max} value of 1.5 seconds based on the worst case RTT. The system bootstraps with arbitrary selected RTT values of 500 ms. Consequently, our algorithm can be ran by considering at each iteration destinations with $RTT < T$ are part of the fast group while destinations with $RTT > T$ as members of the slow group.

B. PMT delays and throughput

By experimenting different capacities for the fast destination (and keeping the same capacity for the other channel), our objective is to investigate the capability of our algorithm of dynamically splitting and merging groups based on the performance of each destination. This is particularly useful when cognitive radios switch to new channels with different capacities and ranges. For this reason, we modify the capacity from 500 Kbits/s to 320 Kbits/s then 250 Kbits/s of the 2.4 Ghz band and show results in Figures 4a, 4b and 4c.

We can observe that when the capacities of the fast destination and the slow destination get closer to each other, the value of T that maximizes the global throughput increases up to a point where only one group is considered (Figure 4c). In Figures 4a and 4b, the value of T that maximizes the global throughput is $T = RTT_{fast} + \epsilon$, and 2 groups are formed: the first one is served every T and the second one every T_{max}.

In Figure 4c, when the gap between the capacities of the two channels is 125 Kbits/s the value of T that maximizes the global throughput is $T = RTT_{slow} + \epsilon$, yielding to a single group served by 4 packets of size 1024 Bytes every T.

These real experiments highlight the ability of PMT to dynamically adapt its group distribution to cope with the varying conditions in tactical point to multipoint context. Moreover, a key feature of the PMT protocol is its capacity to bootstrap from any value of T then smoothly converge to a T value that renders optimal throughput for each group.

We further compare in Figure 5 the estimated overall throughput with our PMT protocol to 2 other potential solutions explained below. Note that these experiments were repeated with the 3 capacities used on the 2.4 Ghz band (500, 320 and 250 Kbits/s).

- **PMT** We serve the fast destination at $T = RTT_{fast} + \epsilon$ and the slow destination at T_{max}, that is the implementation of our PMT protocol.
- **NORM** We serve both destinations at $T = RTT_{slow} + \epsilon$. i.e. T value is based on the slowest receiver’s RTT.
- T_{max} We serve both destinations at T_{max}, that can be considered as the worst case benchmark.

The Figure 5 results are based on the mean T value for the 1 MegaBytes file transfer. As expected, our protocol clearly offers better results than the worst case. More interestingly, PMT outperforms NORM in the case where two groups are formed (500 and 320 Kbits of the 2.4 Ghz band). Quite logically, it offers the same throughput as NORM when a single group is formed (250 Kbits capacity). In this particular case, only one group is formed and is based on the slowest receiver’s performance. Note here that gains would have been even more important if real tactical VHF and UHF were considered, however we have chosen for the sake of simplicity and feasibility of our experiments. In summary, PMT adapts

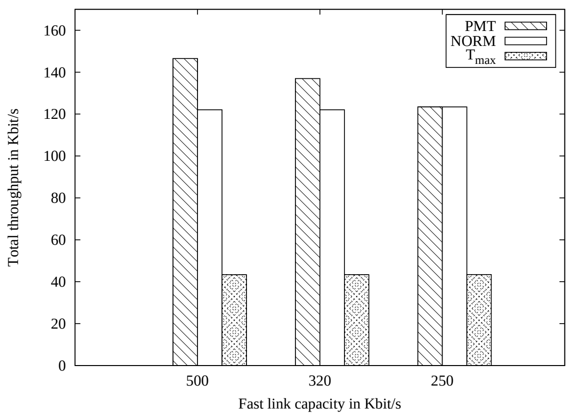


Fig. 5: Global Throughput vs T value.

dynamically to the channel selection, optimizing the global throughput in every case. Indeed, whenever a residual capacity is available on a band, our protocol exploits this valuable resource to offer higher throughput to capable receivers, this results in higher overall network throughput.

V. Related Work

Recently proposed transport protocols for point to point cognitive radio networks [12], [13] do not address challenges of the point-to-multipoint communication scheme. In fact, these solutions make the comprehensive assumption that at time t a single destination needs to be reached. Hence rate adaptation is based on optimizing the transmission parameters based on this destination reception capabilities. Alternatively, present point-to-multipoint transport solutions do not cope well with the new conditions created by tactical environments. Standard multicast solutions target essentially multicast sessions with large groups [14], [8]. For receivers with different flow rates, one can compute multicast groups based on throughput [15] or create layered multicast protocols [16]. They apply well to layered content/stream distribution, where each quality layer is mapped to the corresponding receiver rate.

In the meantime an increasing interest for the GNU Radio development kit have been observed in the wireless research community. Clearly, validating technical contributions in realistic environments hence going beyond simulations is highly encouraged in wireless communication and networking communities. This tendency is gaining momentum with the proliferation of low cost and highly performant off the shelf configurable devices. However, most of the work in the GNU Radio community focuses on physical layer [17] and MAC layer aspects [18], [19]. Very few efforts have looked on implementing higher layer solutions and protocols. In a sense, we are also using GNU Radio as a mean to implement our protocol and test its behavior in real-life conditions, nevertheless our approach, using GNU Radio in order to implement a point to multipoint transport protocol, is to the best of our knowledge one of the first initiatives towards this direction.

VI. Conclusion and future work

In this paper we have presented an acknowledgement based transport protocol for point-to-multipoint next generation tactical networks. Our protocol splits receivers into groups, each served at a suitable throughput thus preventing slow receivers from affecting the service offered to destinations possessing better conditions. We have validated our protocol over a cognitive radio platform. Our experiments prove that our solution hides channel switching of cognitive radios to upper layer. Therefore, our transport protocol is particularly useful to push for new group services such as Blue Force Tracking and multimedia content delivery over emerging tactical radio devices and waveforms.

In the future we plan to extend the solution to N groups. Intuitively, this can be seen as running the same algorithm recursively on the created groups. However, optimality of this solution should be verified. Moreover, optimizing dynamically the T_{max} value is to be considered.

References

[1] B. H. J. Burbank, P. Chimento and W. Kasch., "Key Challenges of Military Tactical Networking and the Elusive Promise of MANET Technology," *Communications Magazine, IEEE*, vol. 44, no. 11, 2006.

[2] "Joint Tactical Radio System Homepage." http://jpeojtrs.mil.

[3] "Le GTIA Renouvelé ,Dossier Scorpion," *Proc. Terre Magazine N 215 - In French*, June 2010.

[4] "ESSOR: European Secure SOftware defined Radio." http://www.occarea.org/28.

[5] R. Iovine and J. Bouis, "The Flexnet-Waveform in the international SDR arena," in *Proc. IEEE MILCOM*, 2009.

[6] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Comput. Netw.*, vol. 50, no. 13, 2006.

[7] B.Adamson, C.Bormann, M. Handley, and M. Handley, "Nack-oriented reliable multicast (norm) transport protocol," *RFC 5740.*, 2009.

[8] S. Paul, K. K. Sabnani, J. C.-H. Lin, and S. Bhattacharyya, "Reliable multicast transport protocol (rmtp)," *IEEE Journal of Seleted Areas in Communication (JSAC)*, vol. 15, no. 3, 1997.

[9] H. Khalife, V. Conan, J. Leguay, and T. Spyropoulos, "Point to multipoint transport in multichannel wireless environments," in *IEEE WCNC conf*, 2013.

[10] "Ettus Research." [Online]. Available: http://www.ettus.com

[11] "The GNU Radio project." [Online]. Available: htpp://gnuradio.org

[12] K. Chowdhury, M. D. Felice, and I. Akyildiz, "Tp-crahn: A transport protocol for cognitive radio ad-hoc networks," in *IEEE infocom conf*, 2009.

[13] C. Luo, F. R. Yu, H. Ji, and V. C. M. Leung, "Optimal channel access for tcp performance improvement in cognitive radio networks," *Springer Wireless Networks*, vol. 17, no. 2, 2011.

[14] G.Kwon and J.Byers, "roma: a reliable overlay multicast with loosely coupled tcp connections," in *IEEE infocom conf*, 2004.

[15] L.Rizzo, "pgmcc: a tcp-friendly single-rate multicast congestion control scheme," in *ACM SIGCOMM conf*, 2000.

[16] G.Kwon and J.Byers, "Smooth multirate multicast congestion control," in *IEEE infocom conf*, 2003.

[17] M. Braun, M. Mller, M. Fuhr, F. K. Jondral, "A USRP-based Testbed for OFDM-based Radar and Communication Systems," *Proc. of 22nd VTech Symposium on Wireless Communications*, 2012.

[18] A. M.-T. A. Puschmann, M. A. Kalil, "A Flexible CSMA based MAC Protocol for Software Defined Radios," *Frequenz.*, vol. 6, no. 9–10, 2012.

[19] G. Nychis, T. Hottelier, Z. Yang, S. Seshan, P. Steenkiste, "Enabling MAC Protocol Implementations on Software-Defined Radios." *NSDI'09 Proc. of the 6th USENIX*, 2012.