

Implementação do protocolo RA-TDMA+ e estudo do impacto da mobilidade

1st Daniel Gonçalves
MIEEC

FEUP - Universidade do Porto
Porto, Portugal
up201402775@fe.up.pt

2nd João Afonso
MIEEC

FEUP - Universidade do Porto
Porto, Portugal
up201403803@fe.up.pt

3rd Paulo Correia
MIEEC

FEUP - Universidade do Porto
Porto, Portugal
up201405129@fe.up.pt

Resumo—Com o objetivo de melhorar a conexão entre *drones* cooperativos que requerem sincronização mútua, é apresentado neste artigo o problema enfrentado, proveniente do uso do protocolo TDMA *standard* nestas situações, a solução pensada e a análise dos resultados após a sua implementação. A solução implementada foi o uso do protocolo RA-TDMA+ desenvolvido pelo Luís Oliveira, Luís Almeida e Daniel Mosse [2]. Por fim, foi analisado a viabilidade desta solução e o impacto de interferências nela para as situações de uso pretendidas.

Palavras-Chave—UDP; CSMA/CA; TDMA; RA-TDMA+; rede *ad-hoc*; wireless; sincronização; redes *mesh* sem fios; relógio global; sincronização sem sinal de relógio; *slots* de tempo separados;

I. DEFINIÇÃO DO PROBLEMA

Drones cooperativos são sistemas de tempo real que requerem sincronização mútua, quer para transmitir dados entre eles, quer para melhorar a qualidade da comunicação *wireless*.

Tendo atenção a este último ponto, uma técnica recorrente é eliminar as colisões de acesso alocando *slots* de tempos predefinidos aos *drones*, sendo esta conhecida como *Time Division Multiple Access* (TDMA). Esta técnica requer o uso de um sinal de relógio global para identificar cada *slot*, no entanto, este método não é robusto o suficiente para transmissões assíncronas geradas por nós externos ou entrada de nós, sendo um dos requisitos que a rede de *drones* seja dinâmica ao ponto de aceitar a entrada e saída de *drones* durante o seu funcionamento normal.

II. DESCRIÇÃO DA SOLUÇÃO

Tendo atenção aos requisitos referidos anteriormente, e também ao facto de que o protocolo seja implementado sobre uma camada de sobreposição (CSMA/CA), surgiu o protocolo RA-TDMA+, em que este sincroniza a comunicação entre *drones* sem o uso de um sinal de relógio global, apenas usando o instante de tempo da receção de pacotes vindos dos *drones* vizinhos. A sincronização atinge-se atrasando o envio do próximo pacote, tendo-se atenção ao atraso máximo recebido de todos os vizinhos na ronda.

A. Ronda de comunicação TDMA:

Os *drones* de uma equipa comunicam um de cada vez num ciclo, a isto chama-se uma ronda de TDMA, a duração desta ronda é definida, fixa e conhecida à priori por todos os *drones* pertencentes à equipa. Em seguida, a ronda é dividida em N consecutivos *slots* com tempo igual a $t_{slot} = T_{up}/N$, sendo cada um atribuído exclusivamente a um *drone* da equipa. Cada *drone* possui um único *ID*, o protocolo usa um simples algoritmo que atribui *slots* aos *drones*, aumentando o seu *ID*. Por exemplo uma equipa constituída por $\{ID_2, ID_4, ID_6\}$, o *drone* ID_2 é lhe designado o *slot* 0, o ID_4 o *slot* 1, e por fim, ID_6 com o último *slot*.

B. Sincronização

No início do *slot* de cada um dos *drones*, este dissemina um pacote com um tipo específico, neste caso *sync*, para os *drones* vizinhos indicando a sua presença. Portanto, sempre que um *drone* recebe um pacote de tipo *sync*, este pode detetar se o pacote foi recebido quando esperado ou atrasado. Se um atraso for detetado, o *drone* atrasa a sua próxima transmissão pelo mesmo valor, ao fim de manter os *slots* em ordem e separados conforme se pode observar através da figura 1. No caso de uma rede *mesh*, o *drone* avalia os atrasos de todos os pacotes *sync* recebidos numa ronda e usa o valor máximo para compensar.

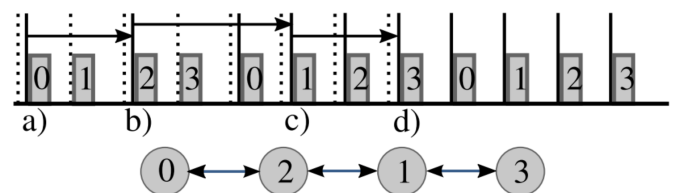


Figura 1. Sincronização RA-TDMA+ : (cima) propagação dos atrasos (baixo) topologia dos *drones* [2]

É de notar que a fonte de onde é proveniente os atrasos é irrelevante para o protocolo, seja vindo de interferência no meio, desvios de relógios internos dos *drones* ou mesmo má configuração dos parâmetros da ronda T_{up} (provocando valores diferentes de duração de *slot* T_{slot}).

O atraso é calculado subtraindo o *current_time* com o tempo inicial ideal do *slot* que enviou o pacote recebido. O tempo de início do *slot* ideal do *drone* que enviou o pacote dá-se por:

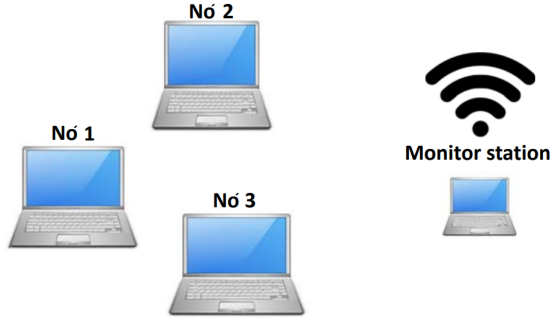


Figura 2. Topologia dos *drones*

$$ideal_slot = begin_ms + slot_width * (sender_id - own_id), \quad (1)$$

onde $begin_ms$ corresponde ao início do slot do *drone* que recebeu o pacote, o $slot_width$ à duração do slot, $sender_id$ ao número do slot do *drone* que enviou o pacote e own_id ao que recebeu este pacote.

Em seguida calcula-se o atraso pela seguinte fórmula:

$$delay = current_time - ideal_slot, \quad (2)$$

Tendo o conhecimento dos atrasos de todos os pacotes recebidos, o valor que é usado para atrasar o início do slot é o valor máximo destes atrasos limitado entre 2% a 25% do valor da duração do slot.

$$delay = max(delay_array[N - 1]) \quad (3)$$

$$begin_ms = begin_ms + delay \quad (4)$$

C. Associação dinâmica da equipa

O protocolo também permite que a composição da equipa altere em emissão, os *slots* podem ser criados ou destruídos dinamicamente, permitindo a entrada e saída de *drones* durante o funcionamento normal. No caso da entrada, cada *drone* analisa o *ID* do pacote recebido e compara com o número de *drones* na equipa, se este *ID* for superior, é dado como uma entrada na equipa.

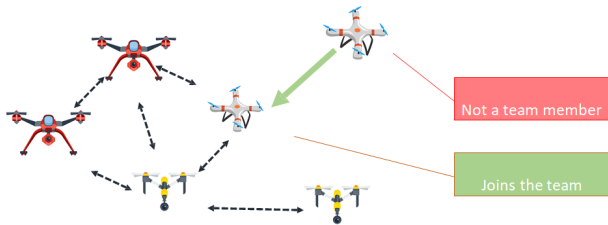


Figura 3. Entrada de um *drone* na equipa

Pelo contrário, na saída de um *drone* foi necessário a criação de um timeout, em que no caso de a não receção de um pacote proveniente de um *drone* consecutivamente até um valor parametrizável, este é assumido como uma saída do sistema. A razão de ter sido necessário implementar este timeout, foi para evitar que uma perda de pacote fosse assumida como uma saída de um *drone* da equipa.

III. RESULTADOS

De forma a analisar os resultados, utilizou-se uma estação de monitorização que observa todos os pacotes que circulam na rede (figura 2). Nas seguintes figuras pode-se analisar a entrada de *drones* na rede, a sua sincronização e posterior saída desse *drone*.

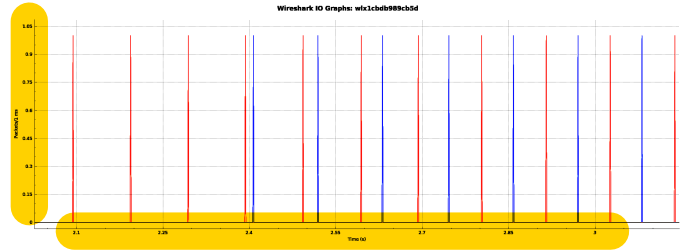


Figura 4. 2 *drones* com duração da ronda de 100 ms.

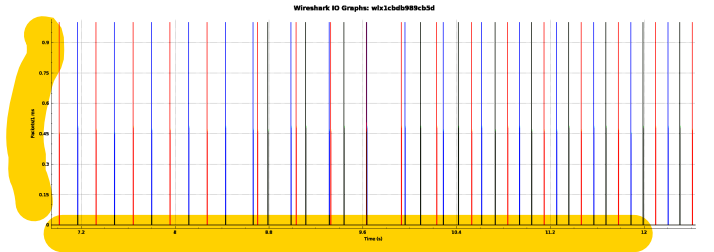


Figura 5. 3 *drones* com duração da ronda de 300 ms.

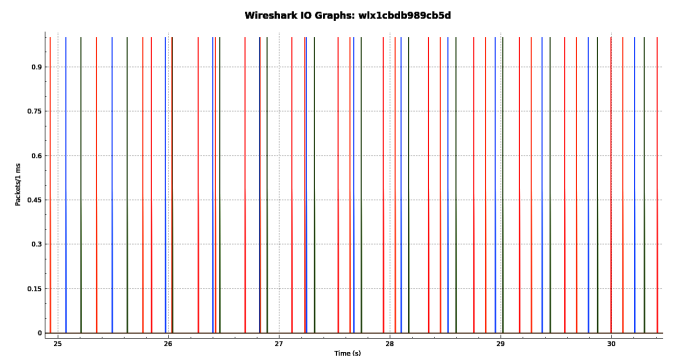


Figura 6. 4 *drones* com duração da ronda de 400 ms.

Na figura 4 podemos inferir que aquando da entrada de um *drone* na rede, estes sincronizam em sensivelmente cerca de quatro rondas, adaptando o tempo alocado a cada *drone* (*slot*) conforme o número destes na rede.

Nas figuras seguintes 5 e 6 evidencia-se o mesmo caso para para três e quatro *drones*, respetivamente.

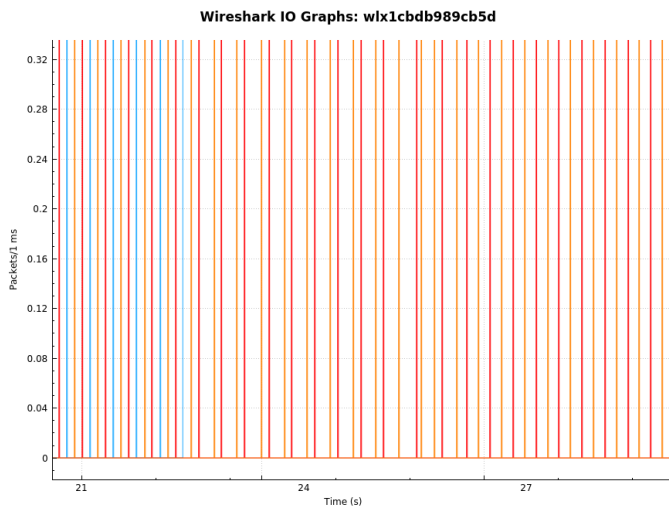


Figura 7. Saída de um *drone*.

A saída ocorre ao fim de um *timeout*, como referido anteriormente. Isto pode ser útil neste tipo de utilizações, pois o *drone* pode necessitar de sair momentaneamente da rede para efetuar alguma operação, retornando prontamente, reutilizando assim a sua *slot* e reduzindo o tempo necessário para a sua nova sincronização com os restantes *drones*.

IV. ANÁLISE DO IMPACTO DA MOBILIDADE

De seguida efetuou-se uma análise sobre o impacto da mobilidade neste protocolo. Na figura 8 foram posicionados os *drones* de forma a que estivessem separados por uma parede e um parcialmente envolvido numa caixa de metal.

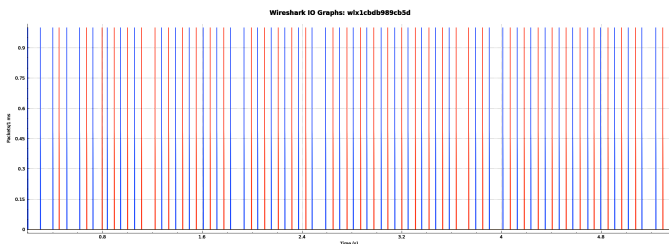


Figura 8. 2 *drones* com duração da ronda de 100 ms com parede e caixa metálica entre eles.

Como se pode observar, estes sincronizaram mesmo existindo algumas perdas de pacotes.

No teste seguinte, foi simulado o movimento de um *drone* e analisou-se o impacto na sincronização destes. Na figura 9 observa-se o início deste teste, onde se junta um *drone* extra e sincroniza-se. De seguida na figura 10 este novo *drone* está se a afastar, sendo observável alguma dessincronização com posterior recuperação desta.

Por fim, na figura 11, observa-se, quando de uma distância significativa entre os *drones*, dificuldade em haver uma boa sincronização, eventualmente todos os pacotes deixam de ser recebidos, que após o timeout definido é dado como uma saída da rede.



Figura 9. 3 *drones* com mobilidade dos mesmos.

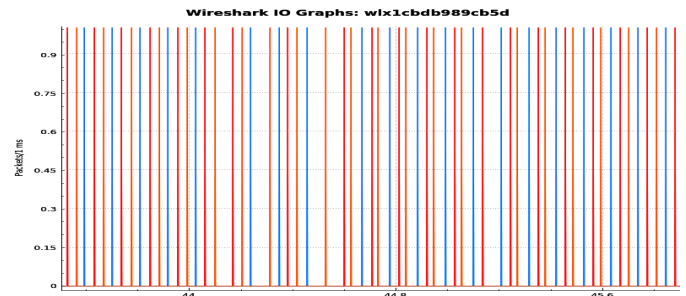


Figura 10. 3 *drones* com mobilidade dos mesmos.

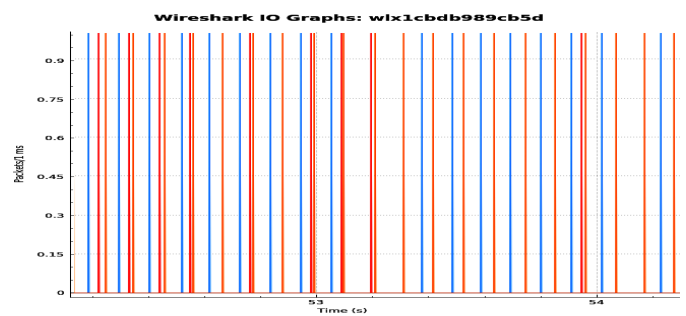


Figura 11. 3 *drones* com mobilidade dos mesmos.

No próximo teste, fomentámos o envio de pacotes extra como interferência, de modo a analisar o impacto que estes têm na sincronização dos *drones*. Como se pode verificar na figura 12, os pacotes a preto foram os de interferência, e em nada afetou a sincronização dos *drones*.

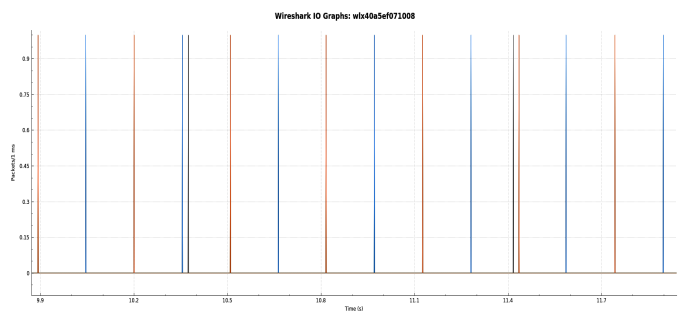


Figura 12. Interferências nas comunicações.

Foram testados, para apenas dois *drones*, a duração mínima de ronda possível. Este parâmetro não é o mais significativo de medir, mas é interessante em alguns casos (*time sensitive*). O melhor caso obtido foi um tempo de ronda de 8ms, onde já se observa alguma dessincronização, sendo esta recuperada na próxima ronda.

Também foram introduzidos erros na parametrização da duração da ronda nos *drones* que se juntavam à rede, observando que estes, caso o erro fosse dentro de 25% do tempo total da ronda, sincronizavam sem problema.

Por fim, foi averiguado que no caso de os *drones* estarem muito afastados, não havia uma sincronização correta, isto deve-se ao facto da existência de um limite máximo de atraso por *drone*, com o aumento da distância entre estes, os atrasos aumentam, sendo necessário o aumento deste limite superior para que ocorra sincronização.

V. ANÁLISE CRÍTICA

Com as condições referidas anteriormente, foi possível concluir que o impacto da mobilidade é significativamente superior ao impacto da distância entre *drones*.

A solução apresentada para resolver o problema foi muito satisfatória, providenciando uma boa sincronização dos *drones* mesmo em casos mais complexos.

Para trabalho futuro, era importante uma análise mais detalhada do impacto da mobilidade dos *drones* na sincronização, com esse intuito para aumentar a eficiência dos testes, o ideal era o uso de um simulador para facilitar a mobilidade dos *drones*.

Para questões de validação da sincronização dos *drones*, era necessário a construção de gráficos mais ilustrativos das variações dos tempos dos *slots* nas rondas.

E por fim, para testar a rapidez de sincronização em número de rondas, era necessário a execução de vários testes, em que o parâmetro do momento da entrada do *drone* na rede influencia fortemente o número de rondas de sincronização, sendo necessário diferenciar os dados destes tempos por entrada na ronda.

VI. ORGANIZAÇÃO DA EQUIPA

No início do projeto, realizou-se um diagrama de Gantt para definir corretamente todas as tarefas necessárias para o cumprimento total do projeto, tendo estas sido divididas em igualdade por todos os membros da equipa. Na parte final, a análise dos resultados e escrita do artigo foi realizada com todos os membros presentes. Posto isso, a nota deve ser distribuída igualmente por todos os membros.

REFERÊNCIAS

- [1] A. Aslam, L. Almeida e F. Santos. «Using RA-TDMA to support concurrent collaborative applications in VANETs». Em: *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*. Jul. de 2017, pp. 896–901. DOI: 10.1109/EUROCON.2017.8011240.
- [2] L. Oliveira, L. Almeida e D. Mosse. «A Clockless Synchronisation Framework for Cooperating Mobile Robots». Em: *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. Abr. de 2018, pp. 305–315. DOI: 10.1109/RTAS.2018.00038.
- [3] L. R. Pinto et al. «Aerial Video Stream over Multi-hop Using Adaptive TDMA Slots». Em: *2017 IEEE Real-Time Systems Symposium (RTSS)*. Dez. de 2017, pp. 157–166. DOI: 10.1109/RTSS.2017.00022.
- [4] João C. G. Reis, Pedro U. Lima e João Garcia. «Efficient Distributed Communications for Multi-robot Systems». Em: *RoboCup 2013: Robot World Cup XVII*. Ed. por Sven Behnke et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 280–291. ISBN: 978-3-662-44468-9.