

Relatório de Pesquisa de Iniciação Científica

Título do Projeto:

Rede de Embarcações Não Tripuladas

Título do Relatório:

Medições de atrasos em Redes
Wi-Fi e ZigBee

UERJ

Faculdade de Engenharia – FEN

Período deste relatório: março de 2017 a julho de 2017

Autor: Ian Henriques de Andrade – Matrícula:201410078211

Professor Orientador: José Paulo Vilela Soares da Cunha

Local de desenvolvimento do projeto: Laboratório de Controle e Automação da UERJ

Rio de Janeiro, 31 de Agosto de 2017

Resumo

Este relatório expõe resultados e conclusões sobre testes realizados para o estudo de atrasos em redes *wireless* dos tipos ZigBee e Wi-Fi, tanto em UDP quanto em TCP. As informações obtidas serão utilizadas para desenvolver e aprimorar o controle de processos com redes sem fio, utilizando o Arduino.

1 Introdução

A proposta deste trabalho foi o estudo de redes sem fio para aplicação em pequenas embarcações não tripuladas (BUORO, 2013; SOUSA, 2016). Anteriormente já havia sido feito um estudo focado na rede ZigBee, a qual possui uma enorme consolidação na área de comunicação sem fio (Callaway et al., 2002) e suas limitações foram discutidas (ANDRADE, 2017), com o qual foi possível obter diversas informações interessantes como estimar a frequência máxima de comunicação, sem haver perdas consideráveis de comandos, por exemplo.

Após isso, foi realizado o estudo da rede Wi-Fi, utilizando o módulo RN-171XV, fabricado pela Roving Networks, o qual pode ser configurado para utilizar TCP ou UDP. A comunicação UDP tem como característica o não estabelecimento de uma conexão entre o remetente e o destinatário, ou seja, é uma comunicação unilateral. Já na comunicação TCP, por outro lado, é previamente estabelecida uma conexão entre o remetente e o destinatário e essa conexão permite haver um mecanismo de controle de envio e recebimento, garantindo a entrega do pacote ao destinatário (KUROSE, 2010).

Foram realizados testes idênticos para cada uma das três opções de comunicação desejada (UDP, TCP e ZigBee) e seus resultados foram coletados e serão expostos neste relatório. Com isso foi possível fazer uma comparação do desempenho entre as Redes ZigBee e Wi-Fi, dessa vez obtendo dados estatísticos sobre ambas, o que será fundamental para a escolha de qual utilizaremos para o prosseguimento do trabalho.

1.1 Objetivo

O objetivo desta etapa do estudo da Iniciação Científica é medir e obter estatísticas do atraso de comunicação sem fio de módulos Wi-Fi e ZigBee conectados ao microcontrolador Arduino.

2 Considerações Iniciais

Os módulos Xbee utilizados são da Série 2 e o software utilizado para configuração foi o X-CTU. As configurações adotadas foram as mesmas relatadas no primeiro relatório.

Os módulos Wi-Fi utilizados são do modelo RN-171XV e os *softwares* utilizados para configuração destes foram o CommOperator e o TeraTerm. Neste foram considerados dois tipos de configurações: TCP e UDP.

O teste realizado consistia no envio periódico de um comando de um módulo Base para um módulo Remoto, ambos acoplados a arduinos UNO e ambos também interligados fisicamente entre duas portas digitais por um fio. Ao enviar um comando, o arduino sinalizava por meio deste elo digital, o temporizador interno do microcontrolador registrava este tempo e ao receber o comando o arduino registrava dessa vez o tempo que o comando foi recebido,

dessa forma é possível obter o tempo de resposta para cada configuração. Para cada configuração o teste foi repetido três vezes, com uma duração de cerca de 1 hora cada teste, para dessa forma obtermos dados suficientes para termos uma estatística mais correta.

Por meio dos registros obtidos serão apresentados, para cada configuração, o número de pacotes perdidos, o maior, menor e a média dos valores de tempo de resposta, o número de vezes que o tempo de resposta ultrapassou 30 ms, a variância dos tempos de resposta, que indica o quão próximo em geral os valores estão da média e um histograma para melhor visualização das distribuições dos tempos de resposta para cada configuração.

3 Sistema de Medição

A aquisição de dados foi realizada por meio do console da IDE do arduino remoto e posteriormente tratados. Os programas utilizados no arduino base, arduino remoto e também no tratamento dos dados obtidos estão no fim do relatório, nos apêndices A, B e C, respectivamente.

Além disso, as Figuras 1 e 2 mostram os diagramas de blocos dos arranjos utilizados nos experimentos, onde o envio/recebimento dos comandos eram realizados pelos módulos Wi-Fi e Zigbee e os pinos 7 e GND de ambos arduinos estavam interligados e serviam para obtermos uma referência temporal e assim ser possível o calculo do tempo de resposta. A sequência cronológica de eventos está demonstrada na Figura 3.



Figura 1- Diagrama de blocos do experimento com módulos ZigBee.

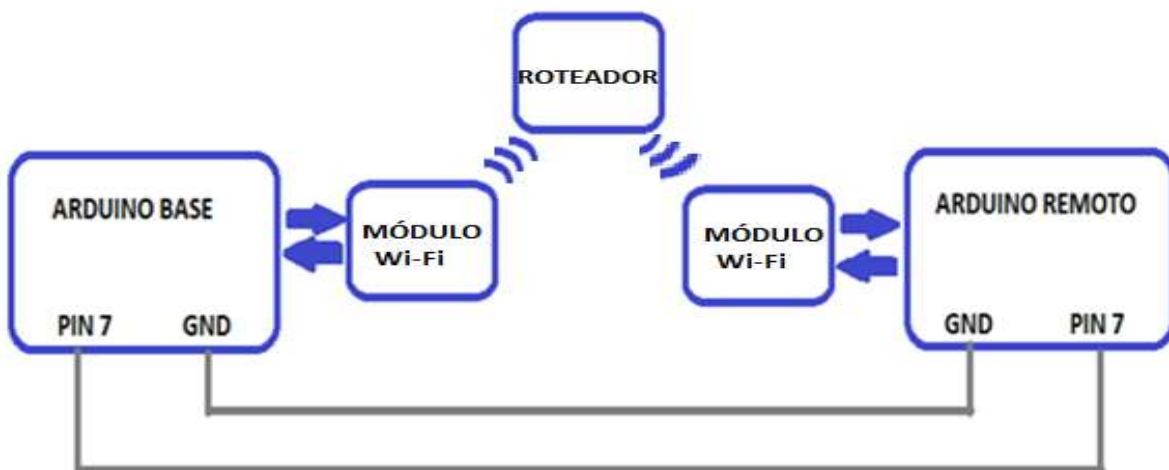


Figura 2 - Diagrama de blocos do experimento com módulos Wi-Fi.

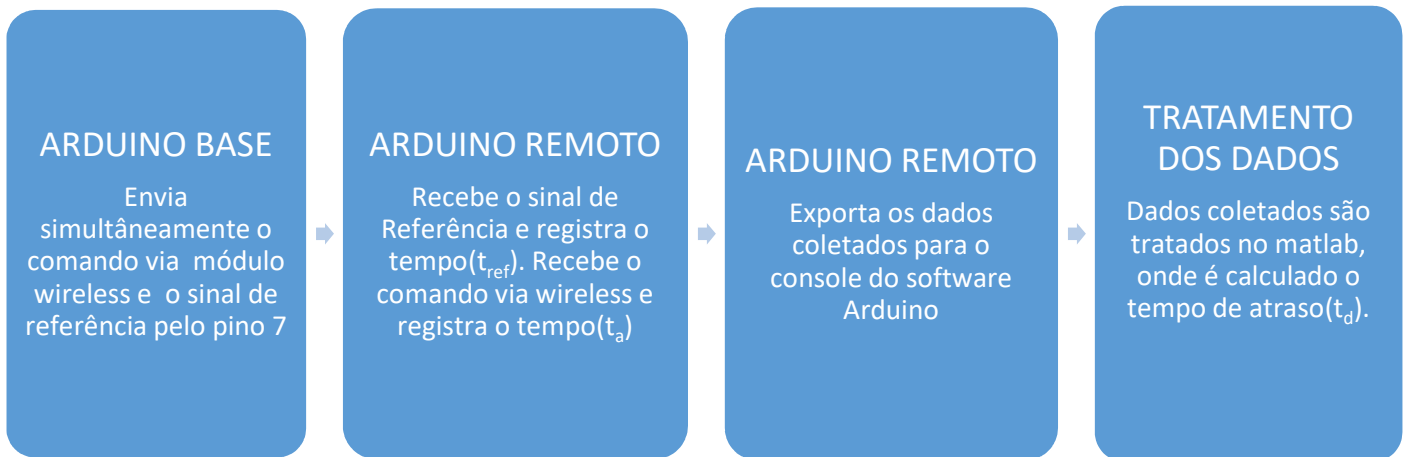


Figura 3 – Sequência de eventos.

O tempo de atraso (t_d) é dado por:

$$t_d = t_a - t_{ref} \quad (1)$$

NOTA: O roteador utilizado foi o TP-LINK N750 modelo TL-WDR4300.

4 Distribuição das Medições dos Tempos de Atraso

O **Módulo Wi-Fi usando UDP** mostrou ser o mais rápido dos testes realizados. É possível observar pelos histogramas da Figura 4 que a variação do tempo de atraso é bem definida em um curto intervalo, que vai de aproximadamente 5,7 ms até 15 ms. Além disso, houve a perda de 117 pacotes dentre aproximadamente 11.000 enviados, ou seja, cerca de 1% dos pacotes foram perdidos.

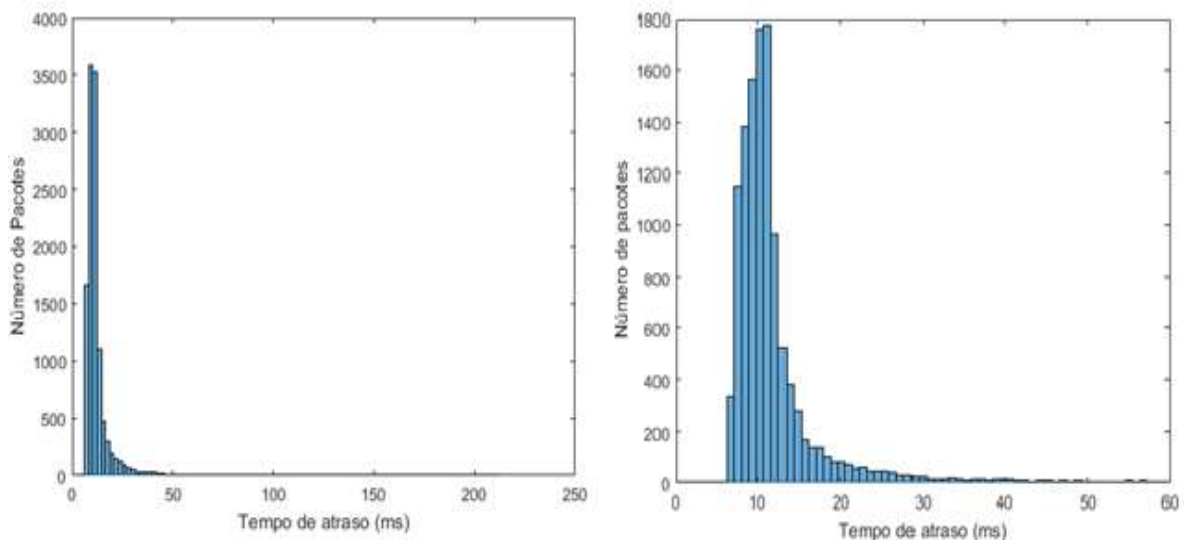


Figura 4 - Histograma completo (Esq.) e ampliado (Dir.) do Tempo de Resposta do **módulo Wi-Fi usando UDP**.

Ao utilizar o **módulo Wi-Fi configurado para TCP**, cuja distribuição dos tempos de atraso está apresentada na Figura 5, percebe-se que possui uma distribuição parecida com o do UDP, porém seu valor médio de tempo de atraso é consideravelmente superior. Nesta configuração não aconteceu perdas de pacotes, porém houve a ocorrência de um “reordenamento” de pacote, o que pode ser esperado pois o TCP promove o reenvio de um pacote até receber a confirmação de que ele foi recebido.

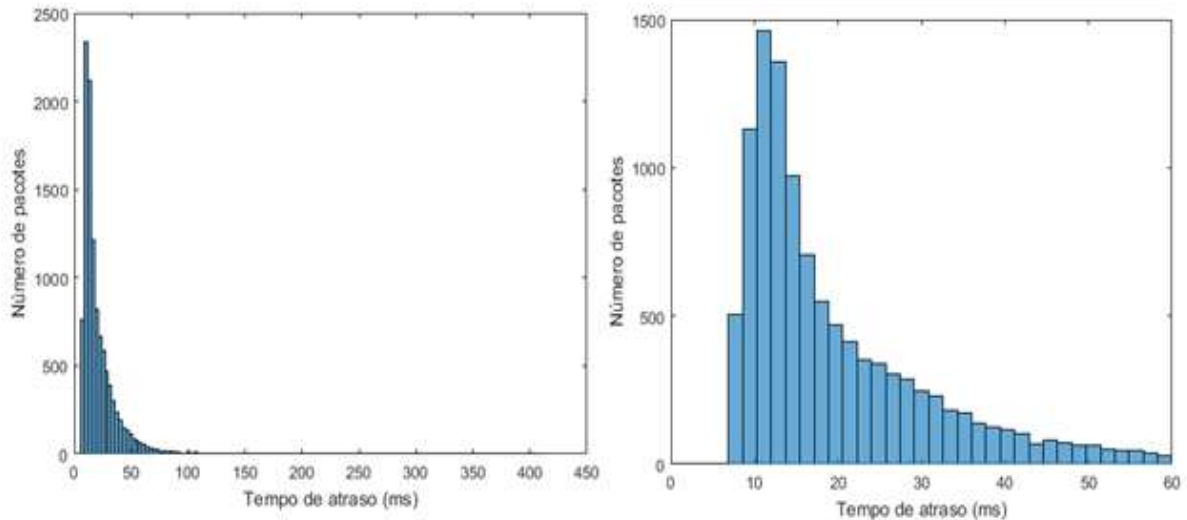


Figura 5 - Histograma completo (Esq.) e ampliado (Dir.) do Tempo de Resposta do **módulo Wi-Fi usando TCP**.

O **módulo Zigbee**, por sua vez, foi o que teve o menor valor médio do tempo de atraso, e também o mais constante, como pode ser observado na Figura 6 (Esq.), além de não haver perdas. Além disso, observando novamente a Figura 6 (Dir.), percebeu-se que sua distribuição de tempo de atraso se concentra em dois intervalos distintos, um em torno de 10 ms e outro próximo de 15 ms, o que parece ser ocasionado pelo sistema de reenvio de pacotes do Zigbee (BEZERRA, 2012), caso estes não sejam reconhecidos.

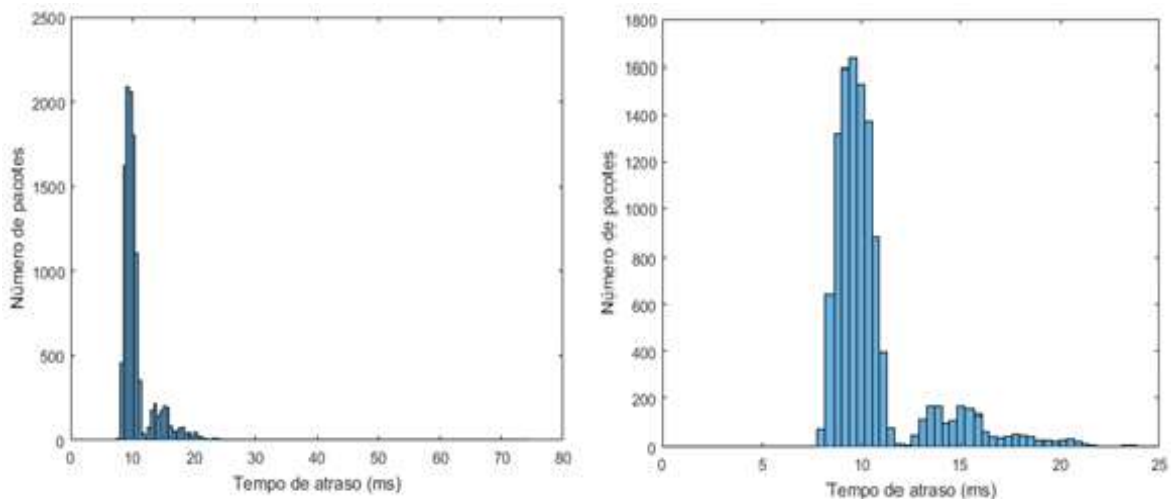


Figura 6 - Histograma completo (Esq.) e ampliado (Dir.) da distribuição do Tempo de Resposta do **módulo ZigBee**.

5 Discussão dos Resultados

Ao observar os dados coletados nos testes apresentados na Seção 4 e na Tabela 1 nota-se que as três configurações possuem diferenças de desempenho consideráveis entre si. Primeiramente analisando a configuração UDP, percebe-se que esta possui o menor tempo de resposta entre elas, porém há consideráveis perdas de pacotes e uma razoável variância do atraso .

Tabela 1 - Dados coletados e Estatísticas para cada configuração.

	Módulo Wi-Fi (UDP)	Módulo Wi-Fi (TCP)	Módulo ZigBee
Total de pacotes	11620	11087	11306
Atraso Mínimo (ms)	5,764	7,080	7,944
Atraso Máximo (ms)	212,124	410,424	74,212
Atraso Médio (ms)	12,465	21,863	10,666
Pacotes Perdidos	117 (1%)	0 (0%)	0 (0%)
Vezes que Ultrapassou 30 ms	347 (3%)	2108 (19%)	5 (0,04%)
Variância do atraso (ms)²	84	358	7,9

Já com a configuração TCP foi notavelmente entre as três aquela com maior variância, conforme a Tabela 1. Isso pode-se atribuir pois nesta configuração o módulo Wi-Fi espera a confirmação do recebimento do pacote e caso isso não ocorra ele reenvia o pacote até obter sucesso. Por isso também, o valor médio dos tempos de resposta é praticamente o dobro das outras duas configurações. Além disso, foi verificado uma ocorrência de o atraso ser maior do que o intervalo entre os envios periódicos dos pacotes.

Por fim, o módulo ZigBee demonstrou ser o mais regular além de não registrar nenhuma perda de comandos no teste. O Valor mínimo de tempo de atraso observado neste caso foi o maior dentre os três, porém seu valor médio é o menor deles, ou seja, pode ser considerado o com melhor desempenho. Além disso, foi observado pela distribuição dos tempos de atraso que o módulo ZigBee possui um mecanismo de reenvio de pacotes, quanto estes não são recebidos, e que este sistema de reenvio de pacotes é muito mais rápido que o do módulo Wi-Fi usando TCP.

6 Conclusão

O estudo e medição dos atrasos em redes sem fio dos tipos ZigBee e Wi-Fi, utilizando o Arduino, foi realizado. A partir disso foi possível conhecer quantitativamente as características de cada tipo de comunicação *wireless* testada, o que será considerado para o desenvolvimento do controle da embarcação que possuímos no Laboratório de Controle e Automação da UERJ.

Após a análise estatística dos dados coletados, é possível inferir que cada configuração tem seus prós e contras. A rede ZigBee pode ser utilizada principalmente para finalidades onde não se admite perdas de pacotes e necessite uma comunicação mais precisa, porém que não precise de uma taxa de comunicação muito elevada, pois, como mencionado anteriormente, essa taxa é limitada pela porta USB do computador.

A rede Wi-Fi utilizando UDP é recomendada quando não há problemas em haver perdas de pacotes e se necessita uma alta taxa de comunicação, pois o envio desses comandos não está atrelado ao envio pela porta USB.

Já a configuração TCP, utilizando o módulo Wi-Fi, pode ser utilizada para projetos que não admitam perdas de pacotes e que não tenha necessidade de um baixo atraso, como por exemplo a monitoração de um sensor de temperatura.

Referências

ANDRADE, I. H., 2017 - **Redes ZigBee e suas limitações**, Relatório de pesquisa de iniciação científica – UERJ.

BEZERRA, J., 2012 – **Instrumentação eletrônica sem fio: transmitindo dados com módulos Xbee ZigBee e PIC16F877A** – 1ª ed. – São Paulo : Érica.

BUORO, A. S., 2013 - **Controle dos Motores e Acionamento de uma pequena Embarcação**, Projeto de Graduação em Engenharia Eletrônica – UERJ.

Callaway, E. ; Gorday, P. ; Hester, L. ; Gutierrez, J. A. ; Naeve, M. ; Heile, B. ; Bahl, V., 2002 - **Home networking with IEEE 802.15.4: a developing standard for low-rate wireless personal area networks** – IEEE Communications Magazine (Volume: 40, Pages 70-77, Issue: 8).

KUROSE, James F., 2010 – **Computer networking fifth edition a top-down approach featuring the internet** (Capítulo 3).

SOUZA, L. R., 2016 - **Acionamento dos Motores CC de uma Embarcação Teleoperada**, Projeto de Graduação em Engenharia Eletrônica – UERJ.

APÊNDICE A - Programa utilizado no Arduino Base

```
int led=13; //Utilizado apenas para visualização
int psinc=7; //Pino que dará o comando de Referência
void setup() {
  pinMode(led,OUTPUT);
  pinMode(7, OUTPUT);
  Serial.begin(115200); //Inicia a comunicação serial à velocidade de 115000
}
void loop() {
  digitalWrite(psinc,HIGH); //Avisa que o pacote 1 será enviado
  Serial.write("1234"); //Envia o Pacote 1
  delayMicroseconds(100);
  digitalWrite(psinc,LOW);
  digitalWrite(led,HIGH);
  delay(1000); // espera X milisegundos
  digitalWrite(psinc,HIGH); //Avisa que o pacote 2 será enviado
  Serial.write("1567"); //Envia o pacote 2
  delayMicroseconds(100);
  digitalWrite(psinc,LOW);
  digitalWrite(led,LOW);
  delay(1000); // espera X milisegundos
}
```


APÊNDICE B - Programa utilizado no Arduino Remoto

```
char valores[4]={}; //vetor onde será armazenado o comando recebido
char liga[4]= {'1','2','3','4'}; //vetor utilizado para comparar comandos recebidos
char desliga[4]={'1','5','6','7'}; //vetor utilizado para comparar comandos recebidos

boolean p=0;

unsigned long time;

void setup() {
  pinMode(13, OUTPUT);
  pinMode(7, INPUT); //Pino utilizado para marcar o tempo de Referência
  Serial.begin(115200); //inicia a comunicação serial à velocidade de 115200 }

void loop() {
  if (digitalRead(7)==HIGH and p==0) //Aguarda o pino de Referência
  { time=micros();

    Serial.print("-"); //Para diferenciar o tempo de referência do tempo de recebimento do
    comando

    Serial.println(time); //Imprime o tempo de referência

    p=1; // garante que só será impresso 1 vez o tempo de referência }

  else if (digitalRead(7)==LOW)//caso o pino de referencia esteja LOW significa que o comando
  já foi enviado {

    p=0; }

  if (Serial.available(>3) { //Espera ter pelo menos 4 caracteres na entrada serial

  time = micros();

  Serial.println(time); // Imprime o tempo que o comando foi recebido

  if(Serial.read()=='1'){ //garante que o primeiro caracter lido seja 1

  valores[0]='1'; //após isso armazena o comando no vetor 'valores'

  for (int i=1; i<4;i++){

  valores[i]=Serial.read(); } }

  if (valores[1]==liga[1] and valores[2]==liga[2] and valores[3]==liga[3]){

  digitalWrite(13,HIGH); }

  if (valores[1]==desliga[1] and valores[2]==desliga[2] and valores[3]==desliga[3]){

  digitalWrite(13,LOW); } } }
```

APÊNDICE C - Programa utilizado para tratamento dos dados no MATLAB

```
n_data = length(data);
data_plus = [];
data_minus = [];
data_loss = 0;
data_disorder = 0;

for i = 1:n_data
    if data(i) >= 0
        if i==n_data
            data_plus = [ data_plus ; data(i) ];
        elseif data(i+1) >= 0
            data_disorder = data_disorder + 1;
        else
            data_plus = [ data_plus ; data(i) ];
        end
    elseif i==n_data
        data_loss = data_loss + 1;
    elseif data(i+1) < 0
        data_loss = data_loss + 1;
    else
        data_minus = [ data_minus ; data(i) ];
    end
end

time_delay = data_plus + data_minus ;

received_packets = length(time_delay)

data_loss
data_disorder
max_time_delay = max(time_delay)
min_time_delay = min(time_delay)
mean_time_delay = mean(time_delay)
var_time_delay = var(time_delay)
n_large_delay = length(find(time_delay>30000))
histogram((time_delay)/1000)
histogram((time_delay)/1000,'BinLimits',[0,60])
```