

Bandwidth dinamico emulado em redes sem fio definidas por software

Mininet-WiFi

Daniel Pereira Monteiro

¹Universidade Federal de Viçosa - Campus Rio Paranaíba (UFV)

Abstract. *This article consists of using the Euclidean distance between a host and a wireless router to build an algorithm that has a greater bandwidth distribution for devices that are towards the end of the wireless network.*

The entire network topology was implemented using the miniet-Wifi emulation tool, which is a mininet fork aimed at emulating virtual access points based on linux wireless drives

Since in wireless networks the user has the possibility to move in different directions and even exceed the limits that the network signal, the distance from the host to the access point is an important factor for sending and receiving packets. therefore, the hybrid approach aims to use the dynamic bandwidth algorithm when the host is moving away from the center of the access point.

Resumo. *Este artigo consiste na utilização da distância euclidiana entre um host e um roteador wireless para a construção de um algoritmo que tenha uma maior distribuição de bandwidth para dispositivos que estão em direção ao fim da rede wireless*

Toda topologia da rede foi implementada utilizando a ferramenta de emulação miniet- Wifi que e um fork do mininet que visa a emulação de pontos de acessos virtuais com base nos drives sem fio do linux

Visto que em redes wireless o usuário tem a possibilidade de se locomover em diferentes direções e até mesmo ultrapassar os limites que o sinal da rede, a distância do host a até o ponto de acesso e um fator importante para o envio e recebimento de pacotes dessa forma a abordagem hibrida visa a utilização do algoritmo de bandwidth dinâmico quando o host está se movimentando em direção oposta ao centro do ponto de acesso.

1. Proposta

Este artigo propõe não somente a utilização de redes sem fio definidas por software para emular a maneira que nos movimentamos e utilizamos redes wireless mas visa a mudança do arquétipo de que a infraestrutura de rede wireless se mantêm estática, este estudo visa então um novo olhar para redes wireless como redes que podem adaptar seus serviços dinamicamente por meio de algoritmos visando assim uma melhor experiência para o usuário

2. trabalhos relacionados

2.1. Mininet Wifi

”Mininet-WiF e um ambiente que permite, a partir de um computador pessoal, criar, explorar e experimentar redes sem fio definidas por software.”[Fontes 2019] retirando

assim a necessidade de se criar toda uma topologia física para realizar os experimentos "O Mininet-WiFi aumenta o conhecido emulador Mininet com estações sem fio virtuais e pontos de acesso, mantendo os recursos SDN originais e a leve arquitetura de software de virtualização. Nós elaboramos as aplicações potenciais do Mininet-Wifi e discutimos os benefícios e as limitações atuais. Dois casos de uso baseados no IEEE 802.11 demonstram a funcionalidade disponível em nossos desenvolvimentos de código aberto."[Fontes et al. 2015]

3. Distancia

Durante o uso de redes wireless é comum utilizarmos dispositivos que permitam que o usuário se mova em diferentes direções e continue conectado a rede, um exemplo disto são os smartphones que utilizam uma determinada rede até que a mesma não seja alcançada pelo smartphone visto que o roteador não consegue prever as movimentações do usuário assumimos que os hosts se movimenta de forma aleatória (movimentação) e gerada por meio Random Walks em 1 dimensão

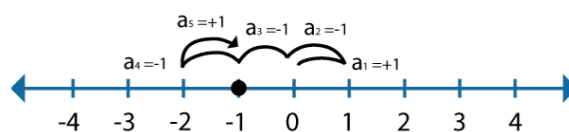


Figura 1. Random Walks 1D

“O passeio aleatório mais simples de entender é um passeio unidimensional. Suponha que o ponto preto abaixo esteja em uma linha numérica. O ponto preto começa no centro. Então, ele dá um passo, para frente ou para trás, com igual probabilidade.”[kardar]



Figura 2. Random Walks 2D
[kardar]

esse modelo de movimentação é implementado por meio da classe do mininet- Wifi

```
1 class RandomDirection(StochasticWalk):
2     def __init__(self, nodes, dimensions, wt_max=None, ...
        border_policy='reflect'):
```

4. demonstração

4.1. topologia

A topologia foi criada utilizando linguagem python o seguinte trecho de código demonstra a criação dos hosts com seus endereços mac criados aleatoriamente e endereçamento de ip para que a rede seja criada emulando elementos de rede

```
1
2 def topology(args):
3
4     net = Mininet_wifi()
5
6     info("*** Creating nodes\n")
7     h1 = net.addHost( 'h1', mac='AA-7F-79-34-99-9D', ...
        ip='10.0.0.1/8' )
8     sta1 = net.addStation( 'sta1', mac='AA-7F-79-34-99-9G', ...
        ip='10.0.0.2/8', range='5' )
9     ap1 = net.addAccessPoint( 'ap1', ssid= 'ap1-ssid', mode= ...
        'g', channel= '1', position='100,100,0', range='80' )
10    c1 = net.addController( 'c1' )
```

4.2. execução

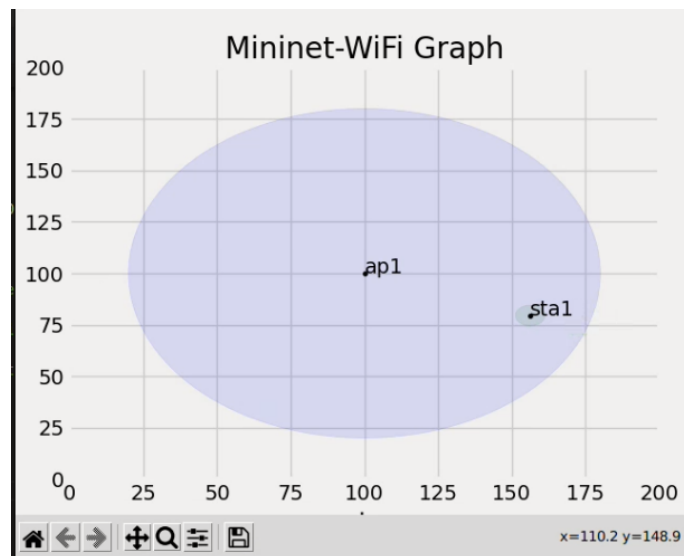
a execução do algoritmo se dá por meio do código

```
1 sudo python3 name.py -V
```

que inicializa a verificação da distância no host ao roteador, depois aloca essa distância em um vetor verificando a cada 2.4 segundos se a distância atual do host aumenta ou não se o host aumenta sua distância, o algoritmo gera um novo bandwidth para que o usuário tenha maior largura de banda quando estiver saindo da rede

```
1 #####
2 #https://github.com/Daniel227a/Bandwidth-dinamico-
3 #emulado-em-redes-sem-fio-definidas-por-software.git
4 #####
5     info("*** Starting network\n")
6     net.build()
7     cl.start()
8     ap1.start([cl])
9     cont=1
10    distan=[]
11    d=-1
12    if '-V' in args:
13        while cont!=25:
14            time.sleep(2.4)
15            x= stal.get_distance_to(ap1)
16            distan.append(x)
17            d=d+1
18            if len(distan)>1 :
19                print("\nDistancia : ",x)
20
21            if len(distan) >1 and (distan[d-1]<distan[d]) and ...
22                (distan[d]<80.0):
23                nodes=net.switches+net.hosts
24                for node in nodes:
25                    for intf in node.intfList(): # loop on ...
26                        interfaces of node
27                        if intf.link: # get link that connects ...
28                            to interface(if any)
29                            newBW = 5+cont
30                            intfs = [ intf.link.intf1, ...
31                                intf.link.intf2 ] #intfs[0] is ...
32                                source of link and intfs[1] is ...
33                                dst of link
34                            intfs[0].config(bw=newBW)
35                            intfs[1].config(bw=newBW)
36                            print("\n")
37
38                cont=cont+1;
```

tal algoritmo pode ser observado por meio do mininet-Wifi graph que mostra o router e o seu raio de abrangência O algoritmo percebe a movimentação e gera um novo bandwidth



se o host estiver dentro do raio da abrangência do roteador wireless que nos teste foi definido como 80 gerando assim um novo bandwidth quando o usuário ainda está dentro do alcance da rede

```

Distancia : 93.79
Distancia : 95.37
Distancia : 103.86
Distancia : 54.53
Distancia : 67.24
(15.00Mbit) (15.00Mbit)
Distancia : 78.84
(16.00Mbit) (16.00Mbit)

```

5. Discussão

Esse experimento abrange um leque de possibilidades gerando várias possibilidades de aprimoramento gerando assim várias questões co a utilização de IA para prever movimentos em redes

6. conclusão

A implementação do algoritmo comprova a viabilidade da implementação e algoritmos que se adaptam as condições do usuário em redes sem fio definidas por software gerando assim a base para a implementação e experimentação desse algoritmo em redes físicas em trabalhos futuros

Referências

Fontes, R. (11 junho 2019). mininet-wifi-ebook-pt-2019121.

Fontes, R. R., Afzal, S., Brito, S. H. B., Santos, M. A. S., and Rothenberg, C. E. (2015). Mininet-wifi: Emulating software-defined wireless networks. In *2015 11th International Conference on Network and Service Management (CNSM)*, pages 384–389.

kardar. [https://www.mit.edu/~kardar/teaching/projects/chemotaxis\(andreaschmidt\)/random.htm](https://www.mit.edu/~kardar/teaching/projects/chemotaxis(andreaschmidt)/random.htm). *mit.edu*.