



**EACH**

Escola de Artes, Ciências e Humanidades  
da Universidade de São Paulo

## **Relatório EP2**

### **Simulação de Subcamada MAC para protocolos do tipo polling**

Alexandre Kenji Okamoto	11208371
Fernanda Cavalcante Nascimento	11390827
Gabriel Afonso Carnaiba Silva	11270886
Karina Duran Munhos	11295911
Mirela Mei	11208392

São Paulo

Neste relatório, foi utilizado um simulador e um artigo como base para as conclusões. O simulador, desenvolvido por terceiros, poderia ser utilizado com outras regras de atendimento, implementando-as na função nextPoll() do programa. Assim como os próprios autores do artigo definiram, para regras que não possuem modelos analíticos que as representem, o uso deste simulador é de suma importância para analisar o comportamento de novos protocolos baseados em polling.

## **Subcamada MAC**

Em modelos de redes compartilhadas típicos, o meio compartilhado deve ser administrado para tornar o transporte eficiente e eficaz. Os mecanismos de acesso ao meio (MAC) são responsáveis por essa administração, evitando conflitos e colisão de pacotes. Esses mecanismos determinam como e quando cada nó utiliza o canal compartilhado e utilizam-se do próprio meio para passar essas informações gerenciadoras.

Para isso, conta com alguns protocolos que precisam atender as necessidades da rede e de aplicação. Dentre eles, tem-se três classes gerais, que são mais eficientes em determinadas características de rede, sendo elas Particionamento de Canal, Acesso Aleatório e Revezamento. Os protocolos de particionamento de canal comumente são por tempo, frequência ou código da rede, enquanto os de acesso aleatório não possui uma coordenação entre os nós a priori e demanda de detecção e correção de colisões. Neste trabalho abordaremos o protocolo de Polling que se enquadra na categoria de Revezamento, que desempenha as maiores vantagens das duas demais categorias.

No caso do protocolo Polling, ele é vantajoso nas situações em que o atraso de propagação de ida e volta dos pacotes é pequeno, a proporção de overhead nos pacotes de mensagens é baixa, cada dispositivo não gera muitos pacotes seguidamente ou a população de dispositivos conectados não é grande.

## **Protocolo Polling**

O protocolo de Polling é do tipo de acesso ao meio livre de disputa. Os dispositivos pertencentes à rede recebem um “convite para transmitir” através de

uma consulta do dispositivo controlador, que determina o envio desses convites de acordo com uma regra de atendimento. Se o dispositivo convidado possuir algo a ser transmitido, ele inicia sua transmissão. Caso contrário, ele envia uma resposta negativa, ou nenhuma resposta (ausência de resposta) que são processadas pelo controlador e ele envia o convite de transmissão ao próximo dispositivo.

As regras de atendimento, que orientam o dispositivo controlador a ordem de envio de convites de transmissão para cada dispositivo, são basicamente quatro:

- Cíclica: O controlador envia convites obedecendo uma ordem cíclica dos dispositivos;
- Aleatória: O controlador escolhe aleatoriamente o próximo dispositivo a ser consultado;
- Elevador: O controlador segue uma ordem para a consulta dos dispositivos e depois de todos consultados, segue a ordem contrária à que foi utilizada;
- Prioridade: o atendimento a cada fila segue uma prioridade pré-definida por alguma função ou parâmetro da rede. Esta regra pode resultar em mais do que uma consulta por ciclo a um determinado dispositivo, ou um tempo maior para o atendimento dos pacotes na fila de determinado dispositivo.

## **Objetivo**

O objetivo do presente relatório é apresentar os resultados obtidos a partir da simulação de subcamada MAC para protocolos do tipo polling, utilizando diferentes regras de atendimento para comparação de seus resultados.

## **Simulação**

### **Código utilizado para Simulação**

Para realizar a simulação, utilizamos dois repositórios disponíveis no GitHub (<https://github.com/phsabo/Polling-Simulator>, e <https://github.com/phsabo/Predict-Polling>, acesso em 26/12/2021) e fomos alterando as variáveis de entrada presentes no código para alterar os ambientes de simulação.

O simulador foi desenvolvido em Python, e implementa um sistema de buffer único com protocolo polling e filas estatisticamente iguais, com intervalos fixos de geração de pacotes para cada fila da rede.

É definido como entrada no código o número de nós da rede, para cada um é definido o intervalo de geração de pacotes, que podem ser fixos ou seguir uma distribuição (p. ex., exponencial, Poisson e normal), e assim cada nó vai gerando e adicionando pacotes na fila.

Como saída temos os seguintes valores:

- Delay/Tempo de espera na fila
- Packets Send
- Lost Packets
- Total Packets
- Success Polls
- Failed Polls
- Total Polls

### **Valores de entrada utilizados para cada Simulação**

- **Simulação 1:**

- Grupo de nós inicial: 1
- Grupo de nós final: 500
- Passo de nós entre cada simulação: 50
- Intervalo entre cada geração de pacotes por nó em mseg: 1000
- Tempo de simulação para cada grupo em minutos: 60
- Distribuição do intervalo de geração de pacotes: Poisson
- Regra de atendimento (Cíclico, Aleatório, Predict, Predict com correção de STD): Cíclico
- Data Rate nominal (250000, 1000000, 2000000): 250000
- Intervalos e distribuição da geração de pacotes dos nós (Aleatório, poisson, exponencial):
- Taxa de sucesso na transmissão dos pacotes de 0-1 (padrão 0.9): 0.9

- **Simulação 2:**

- Grupo de nós inicial: 1
- Grupo de nós final: 500
- Passo de nós entre cada simulação: 50
- Intervalo entre cada geração de pacotes por nó em mseg: 1000
- Tempo de simulação para cada grupo em minutos: 60
- Regra de atendimento (Cíclico, Aleatório, Predict, Predict com correção de STD): Aleatório
- Data Rate nominal (250000, 1000000, 2000000): 250000
- Intervalos e distribuição da geração de pacotes dos nós (Aleatório, poisson, exponential): Poisson
- Taxa de sucesso na transmissão dos pacotes de 0-1 (padrão 0.9): 0.9

- **Simulação 3:**

- Grupo de nós inicial: 1
- Grupo de nós final: 500
- Passo de nós entre cada simulação: 50
- Intervalo entre cada geração de pacotes por nó em mseg: 1000
- Tempo de simulação para cada grupo em minutos: 60
- Regra de atendimento (Cíclico, Aleatório, Predict, Predict com correção de STD): Predict
- Data Rate nominal (250000, 1000000, 2000000): 250000
- Intervalos e distribuição da geração de pacotes dos nós (Aleatório, poisson, exponential): Poisson
- Taxa de sucesso na transmissão dos pacotes de 0-1 (padrão 0.9): 0.9

**Resultados (Presentes em arquivos .csv):**

**Simulação 1:**

```

Polling Simulator
***use -h, --help
Tempo ocioso aguardando poll: 0.7548
time Access Queue: 3.1416
time Service: 2.7743999999999995
limiar STD (listMediaIntervalos/InterMed)*1400
Simulating
20220103-22:42:44
Group Ndes begn      1
Group Ndes end       500
Nodes step           50
Nodes intervalo      1000 mseg
time simulation       60.0 minutos por grupo
Regra de atendimento Ciclico
AirDataRate          250.0 kbps
-- Simul 1 Nodes      |3600.0 segs  |Ciclico      | tempo de sim. 00-00:00:00
mean Delay: 0.000 ms   | success polls: 0.31% | packets send: 3534
lost msgs: 0 0.00%    | tempo 00-00:00:20   | fim em:
-----
-- Simul 51 Nodes     |3600.0 segs  |Ciclico      | tempo de sim. 00-00:00:20
mean Delay: 91.055 ms  | success polls: 17.71% | packets send: 175469
lost msgs: 0 0.00%    | tempo 00-00:00:42   | fim em:20220103-23:03:17
-----
-- Simul 101 Nodes    |3600.0 segs  |Ciclico      | tempo de sim. 00-00:01:02
mean Delay: 215.972 ms | success polls: 41.76% | packets send: 349586
lost msgs: 0 0.00%    | tempo 00-00:01:05   | fim em:20220103-23:04:00
-----
-- Simul 151 Nodes    |3600.0 segs  |Ciclico      | tempo de sim. 00-00:02:08
mean Delay: 402.806 ms | success polls: 77.79% | packets send: 528400
lost msgs: 0 0.00%    | tempo 00-00:01:42   | fim em:20220103-23:08:01
-----
-- Simul 201 Nodes    |3600.0 segs  |Ciclico      | tempo de sim. 00-00:03:50
mean Delay: 678.535 ms | success polls: 94.60% | packets send: 590617
lost msgs: 108358 18.35% | tempo 00-00:02:38   | fim em:20220103-23:13:31
-----

```

```

-- Simul 251 Nodes      |3600.0 segs |Ciclico      | tempo de sim. 00-00:06:28
mean Delay: 969.393 ms  | success polls: 94.62% | packets send: 590681
lost msgs: 278463 47.14% | tempo 00-00:02:35 | fim em:20220103-23:09:08
-----
-- Simul 301 Nodes      |3600.0 segs |Ciclico      | tempo de sim. 00-00:09:03
mean Delay: 1263.162 ms | success polls: 95.10% | packets send: 592308
lost msgs: 453007 76.48% | tempo 00-00:02:19 | fim em:20220103-23:05:23
-----
-- Simul 351 Nodes      |3600.0 segs |Ciclico      | tempo de sim. 00-00:11:23
mean Delay: 1558.429 ms | success polls: 94.84% | packets send: 591429
lost msgs: 629468 106.43% | tempo 00-00:02:16 | fim em:20220103-23:03:58
-----
-- Simul 401 Nodes      |3600.0 segs |Ciclico      | tempo de sim. 00-00:13:39
mean Delay: 1849.884 ms | success polls: 94.96% | packets send: 591849
lost msgs: 800537 135.26% | tempo 00-00:02:40 | fim em:20220103-23:04:39
-----
-- Simul 451 Nodes      |3600.0 segs |Ciclico      | tempo de sim. 00-00:16:20
mean Delay: 2143.757 ms | success polls: 94.86% | packets send: 591492
lost msgs: 974430 164.74% | tempo 00-00:03:10 | fim em:20220103-23:05:21
-----
20220103-230216
Qtde de simulações= 9.98
Tempo de simulação= 00-00:19:31
['nodes', 'delay (ms)', '% lost msgs', 'success polls', 'failed polls', 'total polls', 'Packets Produced', 'Packets Send', 'Tpolls']
['1', '0,0', '0,0', '3534', '1139258', '1142792', '3534', '3534', '1142792']
['51', '91,05529844890124', '0,0', '175469', '815485', '990954', '175471', '175469', '990954']
['101', '215,97170171718602', '0,0', '349586', '487602', '837188', '349607', '349586', '837188']
['151', '402,80574683222744', '0,0', '528400', '150875', '679275', '528459', '528400', '679275']
['201', '678,5350726732927', '18,346576546222003', '590617', '33713', '624330', '699093', '590617', '624330']
['251', '969,393461635073', '47,142704776351366', '590681', '33593', '624274', '869313', '590681', '624274']
['301', '1263,1621977873929', '76,48166156796802', '592308', '30528', '622836', '1045530', '592308', '622836']
['351', '1558,429195935361', '106,43171031518577', '591429', '32183', '623612', '1221163', '591429', '623612']
['401', '1849,8841771050247', '135,26034512181317', '591849', '31393', '623242', '1392705', '591849', '623242']
['451', '2143,7565302090734', '164,74102777383294', '591492', '32066', '623558', '1566291', '591492', '623558']

```

## - Simulação 2:

```

Polling Simulator
***use -h, --help
Tempo ocioso aguardando poll: 0.7548
time Access Queue: 3.1416
time Service: 2.774399999999995
limiar STD (listMediaIntervalos/InterMed)*1400
Simulating
20220103-23:19:00
Group Ndes begn      1
Group Ndes end       500
Nodes step           50
Nodes intervalo      1000 mseg
time simulation       60.0 minutos por grupo
Regra de atendimento random
AirDataRate          250.0 kbps
-- Simul 1 Nodes      |3600.0 segs |random | tempo de sim. 00-00:00:00
mean Delay: 0.000 ms  | success polls: 0.32% | packets send: 3622
lost msgs: 0 0.00%   | tempo 00-00:00:28 | fim em:
-----
-- Simul 51 Nodes     |3600.0 segs |random | tempo de sim. 00-00:00:28
mean Delay: 181.743 ms | success polls: 17.78% | packets send: 176063
lost msgs: 745 0.42%  | tempo 00-00:00:53 | fim em:20220103-23:42:47
-----
-- Simul 101 Nodes    |3600.0 segs |random | tempo de sim. 00-00:01:22
mean Delay: 419.527 ms | success polls: 36.88% | packets send: 318786
lost msgs: 33248 10.43% | tempo 00-00:01:14 | fim em:20220103-23:42:04
-----
-- Simul 151 Nodes    |3600.0 segs |random | tempo de sim. 00-00:02:36
mean Delay: 688.590 ms | success polls: 50.26% | packets send: 398899
lost msgs: 123203 30.89% | tempo 00-00:01:31 | fim em:20220103-23:41:07
-----
-- Simul 201 Nodes    |3600.0 segs |random | tempo de sim. 00-00:04:08
mean Delay: 970.258 ms | success polls: 59.86% | packets send: 448731
lost msgs: 249437 55.59% | tempo 00-00:01:44 | fim em:20220103-23:40:03
-----

```



```

-----
-- Simul 251 Nodes      |3600.0 segs |random      | tempo de sim. 00-00:05:53
mean Delay: 1258.514 ms | success polls: 65.75% | packets send: 476674
lost msgs: 395340 82.94% | tempo 00-00:02:15 | fim em:20220103-23:41:52
-----
-- Simul 301 Nodes      |3600.0 segs |random      | tempo de sim. 00-00:08:08
mean Delay: 1547.658 ms | success polls: 70.57% | packets send: 498207
lost msgs: 550614 110.52% | tempo 00-00:02:26 | fim em:20220103-23:41:12
-----
-- Simul 351 Nodes      |3600.0 segs |random      | tempo de sim. 00-00:10:34
mean Delay: 1838.800 ms | success polls: 73.43% | packets send: 510435
lost msgs: 709377 138.97% | tempo 00-00:02:41 | fim em:20220103-23:41:12
-----
-- Simul 401 Nodes      |3600.0 segs |random      | tempo de sim. 00-00:13:15
mean Delay: 2128.440 ms | success polls: 75.90% | packets send: 520718
lost msgs: 870823 167.24% | tempo 00-00:03:01 | fim em:20220103-23:41:35
-----
-- Simul 451 Nodes      |3600.0 segs |random      | tempo de sim. 00-00:16:17
mean Delay: 2420.773 ms | success polls: 77.81% | packets send: 528504
lost msgs: 1033226 195.50% | tempo 00-00:03:08 | fim em:20220103-23:41:31
-----
20220103-233827
Qtde de simulações= 9.98
Tempo de simulação= 00-00:19:27
['nodes', 'delay (ms)', '% lost msgs', 'success polls', 'failed polls', 'total polls', 'Packets Produced', 'Packets Send', 'Tpolls']
['1', '0,0', '0,0', '3622', '1139093', '1142715', '3622', '3622', '1142715']
['51', '181,74321149963916', '0,4231439882314853', '176063', '814366', '990429', '176816', '176063', '990429']
['101', '419,5273555023946', '10,429567170452906', '318786', '545603', '864389', '352076', '318786', '864389']
['151', '688,590047033703', '30,885763062830442', '398899', '394741', '793640', '522181', '398899', '793640']
['201', '970,2578996813118', '55,587200349429835', '448731', '300900', '749631', '698285', '448731', '749631']
['251', '1258,5144581917796', '82,93718558176027', '476674', '248281', '724955', '872191', '476674', '724955']
['301', '1547,6581716738374', '110,51912156994985', '498207', '207731', '705938', '1049050', '498207', '705938']
['351', '1838,7998253797325', '138,97499191865762', '510435', '184706', '695141', '1220089', '510435', '695141']
['401', '2128,4401113560234', '167,2350485291463', '520718', '165341', '686059', '1391862', '520718', '686059']
['451', '2420,772890665121', '195,50012866506214', '528504', '150679', '679183', '1562106', '528504', '679183']
PS C:\Users\nanda\OneDrive\Documentos\each_docs\6\REDES\codigo_simulacao>

```

### - Simulação 3:

```

Polling Simulator
***use -h, --help
Tempo ocioso aguardando poll: 0.7548
time Access Queue: 3.1416
time Service: 2.7743999999999995
limiar STD (listMediaIntervalos/InterMed)*1400
Simulating
20220104-20:31:11
Group Ndes begn      1
Group Ndes end       500
Nodes step           50
Nodes intervalo      1000 mseg
time simulation       60.0 minutos por grupo
Regra de atendimento predict
AirDataRate          250.0 kbps
-- Simul 1 Nodes      |3600.0 segs   |predict      | tempo de sim. 00-00:00:00
mean Delay: 0.000 ms   | success polls: 3.19% | packets send: 3225
lost msgs: 0 0.00%    | tempo 00-00:04:21   | fim em:
-----
-- Simul 51 Nodes     |3600.0 segs   |predict      | tempo de sim. 00-00:04:21
mean Delay: 6.814 ms   | success polls: 23.78% | packets send: 177988
lost msgs: 0 0.00%    | tempo 00-00:04:33   | fim em:20220104-21:28:41
-----
-- Simul 101 Nodes    |3600.0 segs   |predict      | tempo de sim. 00-00:08:54
mean Delay: 23.759 ms  | success polls: 45.43% | packets send: 349993
lost msgs: 9 0.00%    | tempo 00-00:03:29   | fim em:20220104-21:11:32
-----
-- Simul 151 Nodes    |3600.0 segs   |predict      | tempo de sim. 00-00:12:24
mean Delay: 396.327 ms | success polls: 82.14% | packets send: 500376
lost msgs: 22967 4.59% | tempo 00-00:03:44   | fim em:20220104-21:13:26
-----
-- Simul 201 Nodes    |3600.0 segs   |predict      | tempo de sim. 00-00:16:09
mean Delay: 679.346 ms | success polls: 93.84% | packets send: 588032
lost msgs: 109182 18.57% | tempo 00-00:03:52   | fim em:20220104-21:14:23
-----

```

```

-----
-- Simul 251 Nodes          |3600.0 segs |predict    | tempo de sim. 00-00:20:01
mean Delay: 972.990 ms      | success polls: 94.70% | packets send: 590969
lost msgs: 283900 48.04%    | tempo 00-00:04:28    | fim em:20220104-21:18:13
-----
-- Simul 301 Nodes          |3600.0 segs |predict    | tempo de sim. 00-00:24:30
mean Delay: 1262.864 ms     | success polls: 94.48% | packets send: 590202
lost msgs: 451623 76.52%    | tempo 00-00:05:37    | fim em:20220104-21:25:00
-----
-- Simul 351 Nodes          |3600.0 segs |predict    | tempo de sim. 00-00:30:08
mean Delay: 1556.837 ms     | success polls: 94.57% | packets send: 590530
lost msgs: 626324 106.06%   | tempo 00-00:04:36    | fim em:20220104-21:19:47
-----
-- Simul 401 Nodes          |3600.0 segs |predict    | tempo de sim. 00-00:34:44
mean Delay: 1851.854 ms     | success polls: 94.73% | packets send: 591055
lost msgs: 804464 136.11%   | tempo 00-00:04:56    | fim em:20220104-21:20:44
-----
-- Simul 451 Nodes          |3600.0 segs |predict    | tempo de sim. 00-00:39:41
mean Delay: 2145.469 ms     | success polls: 94.85% | packets send: 591480
lost msgs: 979789 165.65%   | tempo 00-00:05:17    | fim em:20220104-21:21:20
-----
20220104-211610
Qtde de simulações= 9.98
Tempo de simulação= 00-00:44:59
['nodes', 'delay (ms)', '% lost msgs', 'success polls', 'failed polls', 'total polls', 'Packets Produced', 'Packets Send', 'Tpolls']
['1', '0,0', '0,0', '3225', '98019', '101244', '3225', '3225', '101244']
['51', '6,814105728542602', '0,0', '177988', '570533', '748521', '177988', '177988', '748521']
['101', '23,758556001078425', '0,002571480001028592', '349993', '420386', '770379', '350002', '349993', '770379']
['151', '396,3266245653205', '4,589948358834157', '500376', '108799', '609175', '523384', '500376', '609175']
['201', '679,3463025731329', '18,567356878537222', '588032', '38581', '626613', '697332', '588032', '626613']
['251', '972,9896843023674', '48,039744893556175', '590969', '33050', '624019', '875031', '590969', '624019']
['301', '1262,8635165652718', '76,52007278863847', '590202', '34495', '624697', '1042044', '590202', '624697']
['351', '1556,837284498419', '106,06133473320577', '590530', '33876', '624406', '1217122', '590530', '624406']
['401', '1851,8536551709299', '136,10645371412136', '591055', '32888', '623943', '1395833', '591055', '623943']
['451', '2145,468614841476', '165,65040238046933', '591480', '32088', '623568', '1571642', '591480', '623568']
PS C:\Users\nanda\OneDrive\Documents\each_docs\6\REDES\codigo_simulacao>

```

## Equações utilizadas no modelo analítico utilizado para a validação do simulador:

Para compreender as equações, é necessário compreender as variáveis utilizadas:

- $N$  é o número de filas (ou dispositivos)
- $\lambda_i$  é a taxa de chegada de pacotes no sistema da fila  $i$
- $b_i$  tempo médio de serviço para atender a fila  $i$

Temos que a utilização do servidor devido à fila  $i$  se dá por:

$$\rho_i = \lambda_i b_i$$

Sendo assim, a utilização total do servidor é a soma de todas as utilizações das  $N$  filas.

$$\rho_T = \sum_{i=1}^N \rho_i$$

$r_i$  é o tempo médio para o servidor trocar de fila. Sendo assim, o tempo total para acessar todas as filas sem executar nenhuma ação é a soma de todos os tempos médios.

$$R = \sum_{i=1}^N r_i$$

Em sistemas simétricos, isto é, quando as filas são idênticas, temos  $R = Nr$ ,  $\rho = \lambda b$  e  $\rho_T = N\rho$ . Neste simulador são utilizadas filas idênticas.

A média de tempo dos ciclos - tempo necessário para que o servidor acesse todas as filas realizando o serviço quando necessário ( $E[C]$ ).

Para buffer único, caso do polling, segundo Takagi (Takagi, H., 1988) :

$$E[C] = \frac{R}{1 - \rho r^b}$$

Média de espera das mensagens na fila ( $E[W]$ )

$$E[W] = \frac{R + \rho r^b}{2(1 - \rho r^b)}$$

### Observações sobre os retornos:

Porcentagem de pacotes perdidos -> Lost messages

Delay mean, ou média de delays, ou  $E[W]$ . Delay, ou tempo de espera, é a diferença entre o tempo atual e o instante em que o pacote entrou no buffer.

O Time service e o Time queue, ou tempo de acesso à fila, são calculados pelo parâmetros da rede fornecidos pelo usuário e as características de rede como, por exemplo, velocidade de transmissão e suas limitações físicas. O tempo de serviço Time Service é obtido subtraindo o tempo de acesso a fila do tempo de uma comunicação completa. Enquanto o Time queue, durante a troca de pacotes, é obtido somando os tempos. Temos também o tempo de comunicação que se dá pela soma do Time queue e o Time service.

## Conclusão:

O trabalho utilizado como base comparou os resultados obtidos por essa simulação com um modelo analítico. Para comparar os resultados obtidos e apresentar os novos resultados, utilizamos de gráficos:

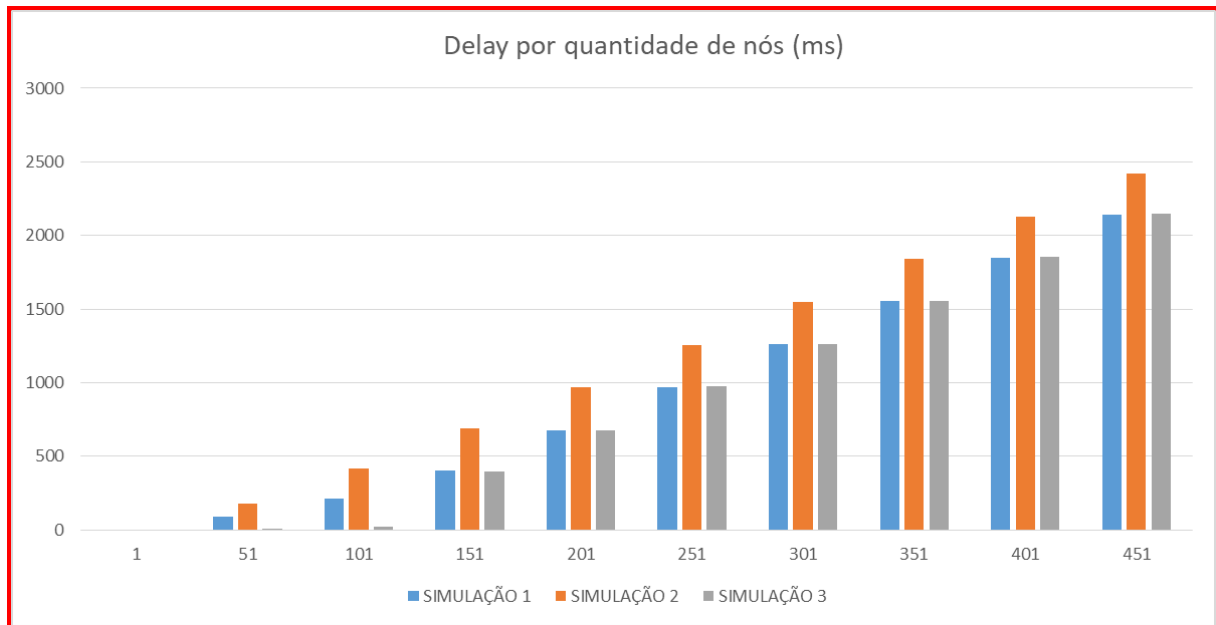


Gráfico 1: Comparação do delay por quantidade de nós entre as simulações.

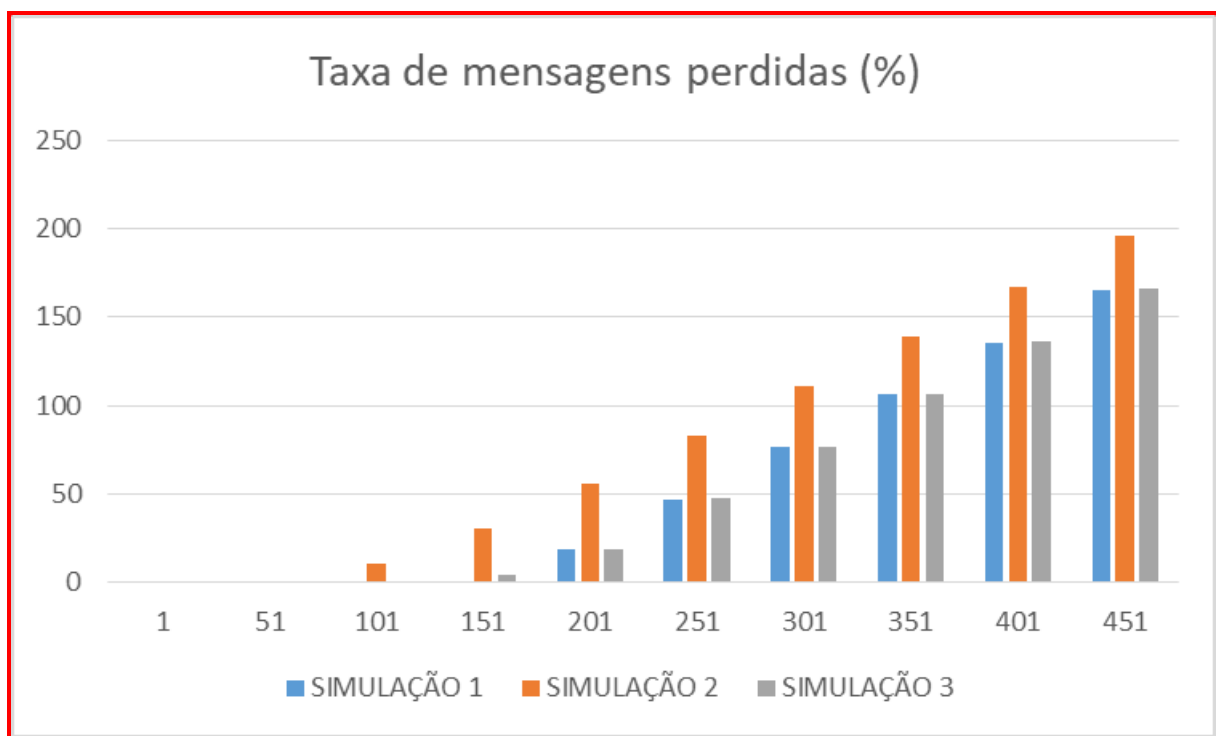


Gráfico 2: Comparação da taxa de mensagens perdidas por quantidade de nós entre as simulações.

A partir dos resultados da simulação, combinando a quantidade de polls por segundo(PpS) enviados com o número de nós N e a taxa de chegada dos pacotes na fila  $\lambda_i$ , podemos estimar quantos polls em média são enviados por pacote e o tempo de acesso a todas as filas  $E[C]$ .

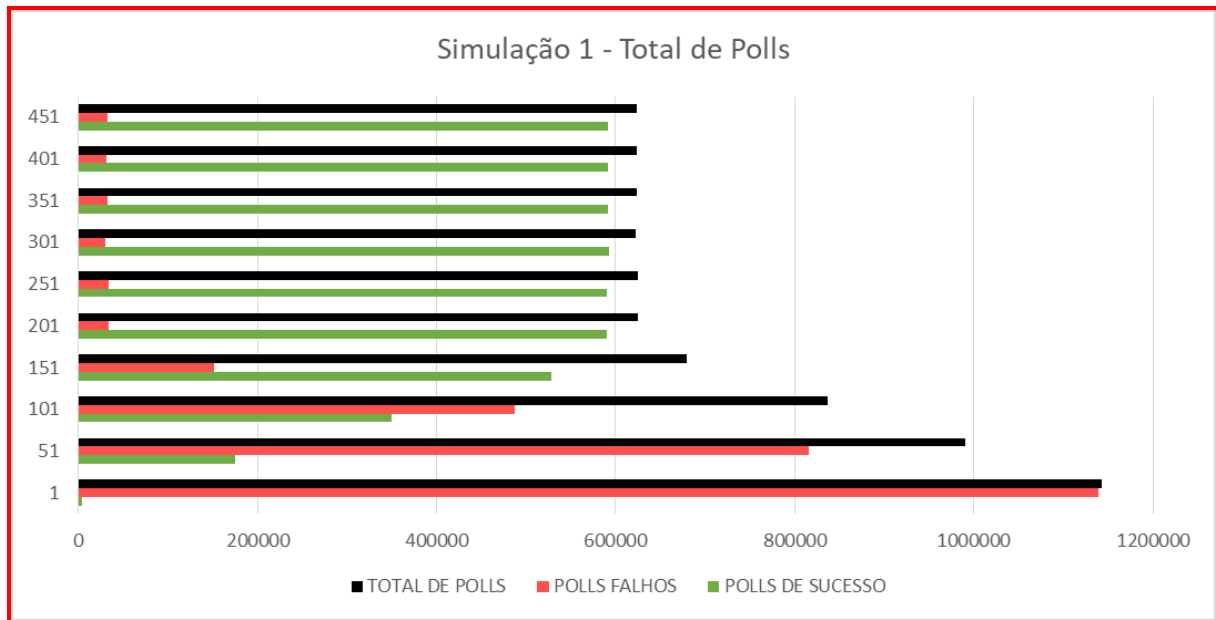


Gráfico 3: Comparação da quantidade de polls falhos e de sucesso por quantidade de nós na Simulação 1.

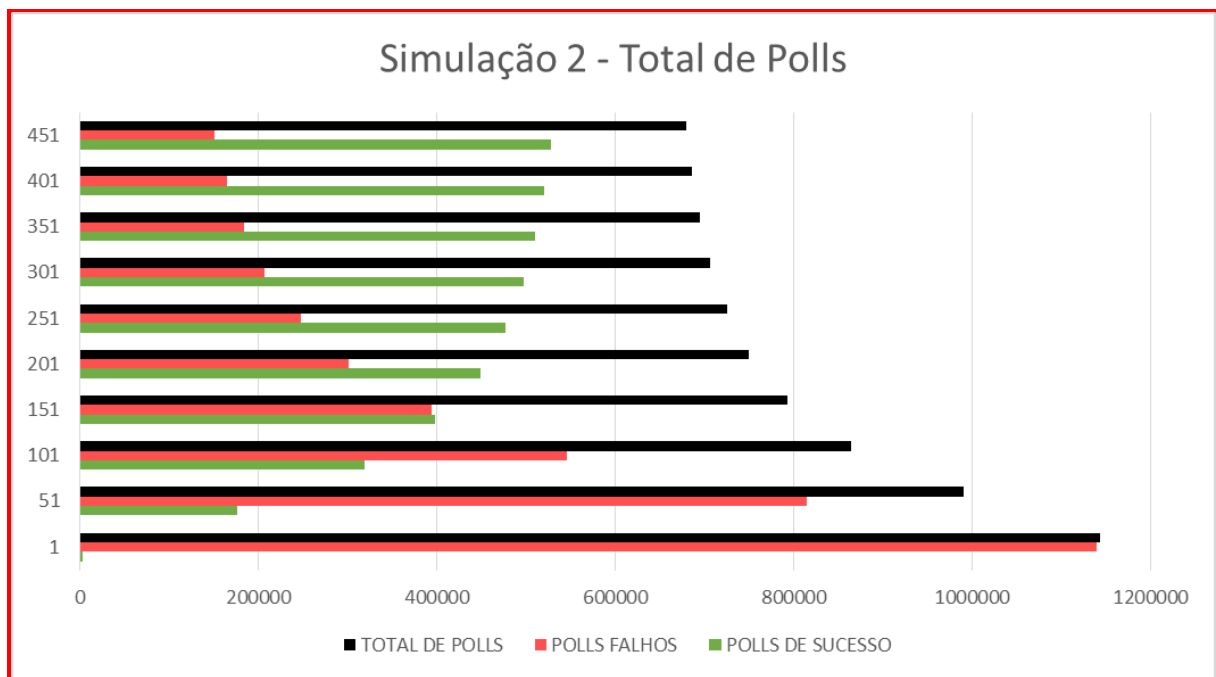


Gráfico 4: Comparação da quantidade de polls falhos e de sucesso por quantidade de nós na Simulação 2.

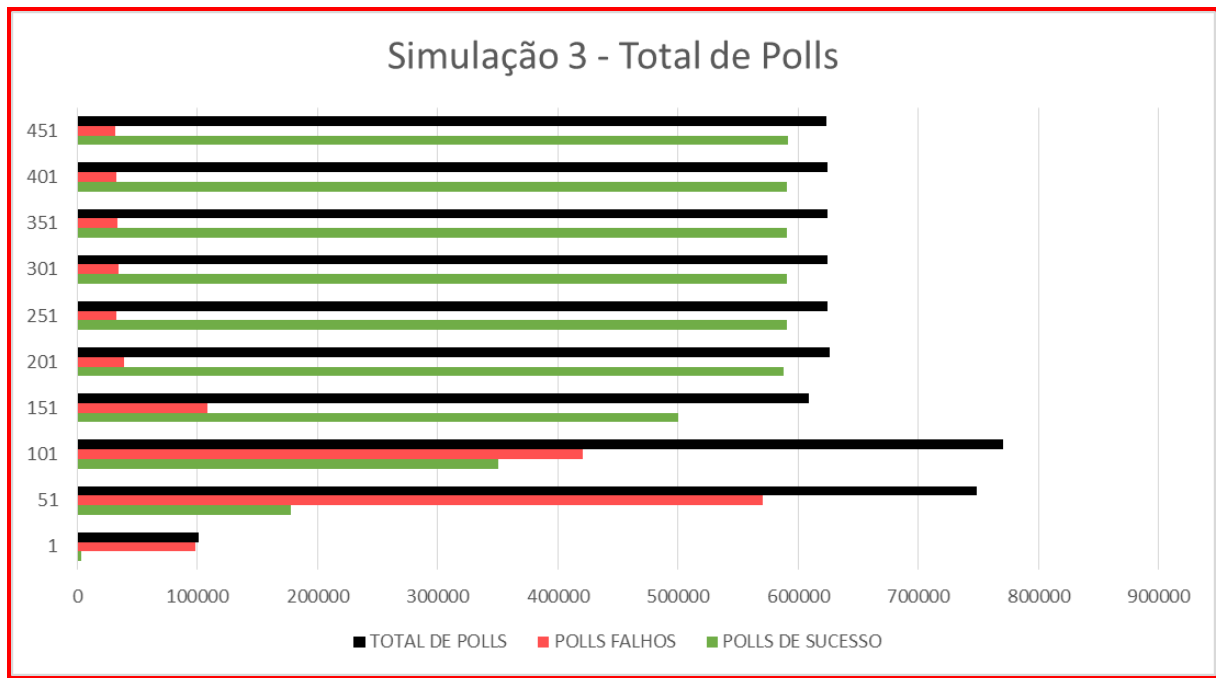


Gráfico 5: Comparação da quantidade de polls falhos e de sucesso por quantidade de nós na Simulação 3.

Para apresentar melhor desempenho, um protocolo de rede deve sempre procurar diminuir qualquer tempo de espera e a quantidade de pacotes perdidos. Assim, buscando a configuração que obteve melhor resultado nestes aspectos, este trabalho expõe as diferenças dos pontos chave.

O objetivo usual na análise de modelos de polling é encontrar o tempo de espera da mensagem. O tempo médio de resposta é uma das medidas de desempenho mais importantes, sendo definido como a soma do tempo médio de espera e do tempo médio de atendimento.

Utilizando gráficos foi possível observar a diferença de desempenho entre as configurações possíveis demonstrando as melhores condições para o polling.

Podemos perceber pelo gráfico 2, que as simulações 1 e 3 se mantiveram muito próximas no tempo de delay durante sua execução, utilizando as regras de atendimento cíclica e predict, respectivamente. Já a regra de atendimento aleatória apresentou um delay consideravelmente maior em sua execução.

É possível admitir, portanto, que dadas as condições de rede e entradas do algoritmo de simulação usado, o pior desempenho foi o da simulação 2, com a regra de atendimento Aleatório. Os resultados das demais simulações são muito parecidos, tendo algumas vezes leve superioridade em algumas métricas de um em

relação ao outro, entretanto de modo geral as regras de atendimento ciclico e predict desempenharam a mesma eficiência.

#### Bibliografia:

Sabo, Paulo Henrique; Cardieri, Paulo. Simulador de Subcamada MAC para protocolos do tipo polling. Alternate title: Iberian Journal of Information Systems and Technologies: Format guidelines to write articles. **Revista Ibérica de Sistemas e Tecnologias de Informação**; Lousada Ed. E28, (Apr 2020): 520-531. Disponível em:

<<https://www.proquest.com/openview/e3b85a7260c758fd0995c5924422ab5b/1?pq-origsite=gscholar&cbl=1006393>>, acesso em: 08 de janeiro de 2022.