Telemetria Adaptativa usando Aprendizado por Reforço Profundo em Redes Definidas por *Software*

Debora Helena Job (UNIRIO)
Sidney Cunha de Lucena (UNIRIO)
Pedro Nuno de Souza Moura (UNIRIO)

5º Workshop de Medições do Comitê Técnico de Monitoramento (CT-Mon) da RNP 2023 – *online*

Sumário da apresentação

- Automação dos processos de gerenciamento de redes
- Telemetria in-band
- Problematização (benefícios x custos)
- Proposta de solução
- Aprendizado por reforço profundo
- Framework SmartMon
- Ambiente experimental
- Resultados
- Conclusões, trabalhos futuros e agradecimentos

Automação dos processos de gerenciamento de redes

Avanços na programabilidade das redes (plano de controle e plano de dados):

- Maior flexibilidade no encaminhamento dos dados
- Gerenciamento centralizado
- Automação de processos
- Monitoramento mais especializado

(P4, eBPF)

Monitoramento especializado + Programabilidade no plano de dados:

- Verificação de problemas em escalas de tempo menores

Telemetria

Telemetria in-band

Dados de telemetria são transmitidos por dentro da rede de dados

- O oposto ao *out-of-band*

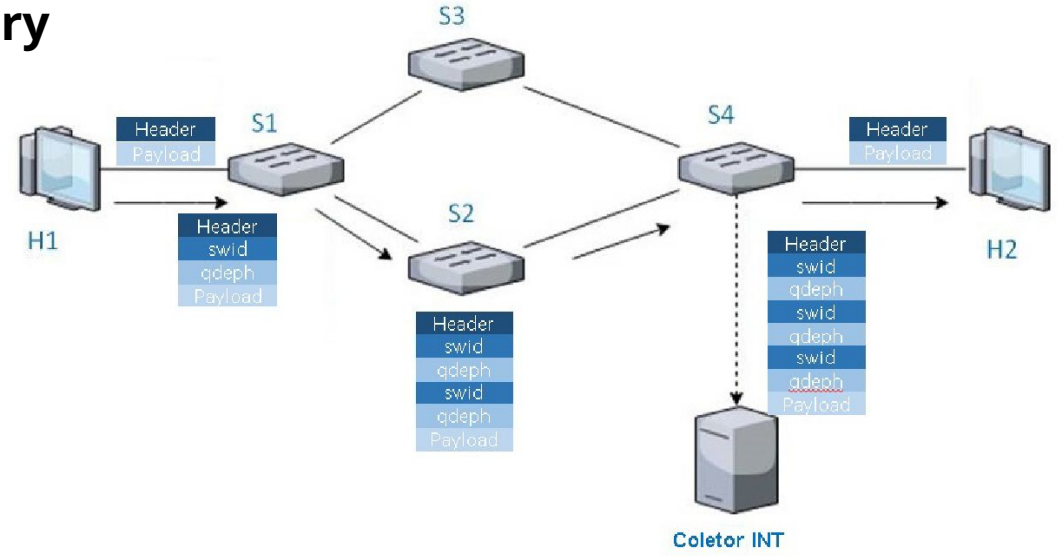
(a mesma que está sendo monitorada)

INT: In-band Network Telemetry

Coletas em **fluxos pré-definidos (fluxos monitores)**

Metadados:

- Switch ID,
- Ingress Port ID, Ingress Timestamp,
- Egress Port ID, Egress Timestamp,
- Hop Latency,
- Egress port TX Link Utilization,
- Queue occupancy



Problematização (benefícios x custos)

Benefícios:

- Riqueza de informações sobre o funcionamento da rede c/ alta granularidade
- Possibilita a construção de datasets p/ ML

AI p/ (semi)automatizar tomadas de decisão

(escalas de tempo minúsculas)

SDN + Telemetria In-band + AI:

- viabiliza o que é chamado de ***Redes Definidas por Conhecimento (KDN)***

Mas a telemetria in-band tem seu custo ...

Problematização (benefícios x custos)

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

Custos (no caso do INT):

- Processamento nos SWs (em especial no SW de ***sink***)

  Depende do HW, nº de fluxos c/ metadados em cada SW etc

- *Overhead* de banda (em especial para pacotes pequenos)

  Pode não ser problema, depende do tipo de tráfego

- Processamento no coletor INT

  Pode não ser problema, dependendo do HW do coletor e do uso de múltiplos coletores distribuídos (*e se estiver localizado na nuvem?*)

- **Altíssimo volume de informações armazenadas no BD do coletor**

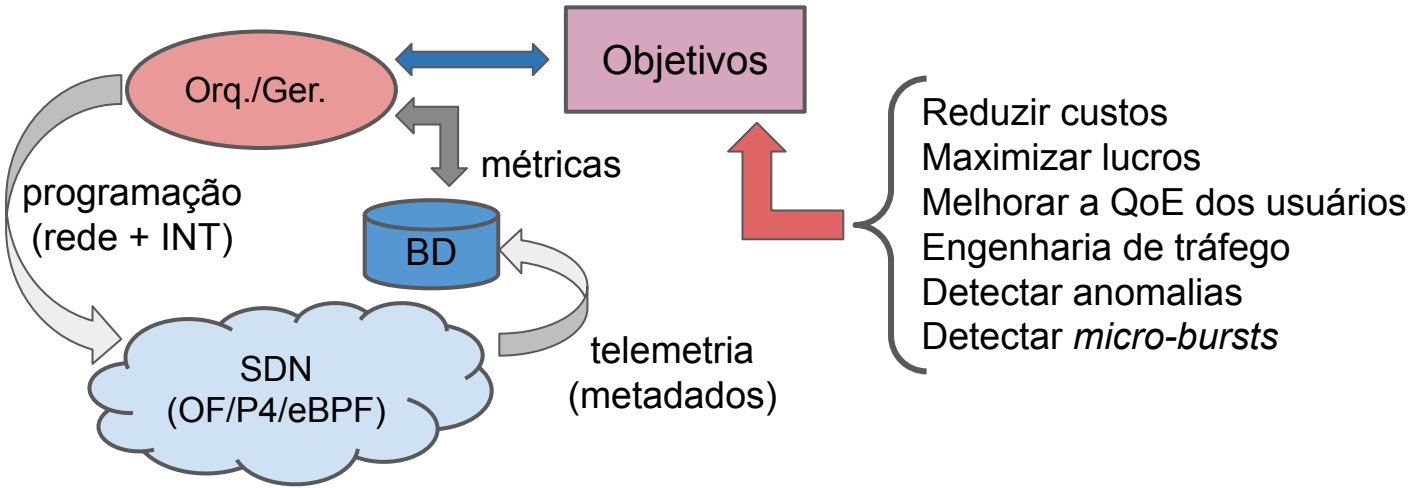
Boa parte pode ser irrelevante ⟵ ⟶ **Dificulta mineração e análises**

Proposta de solução

Telemetria in-band (INT) adaptativa provida por IA

usando aprendizado por reforço profundo

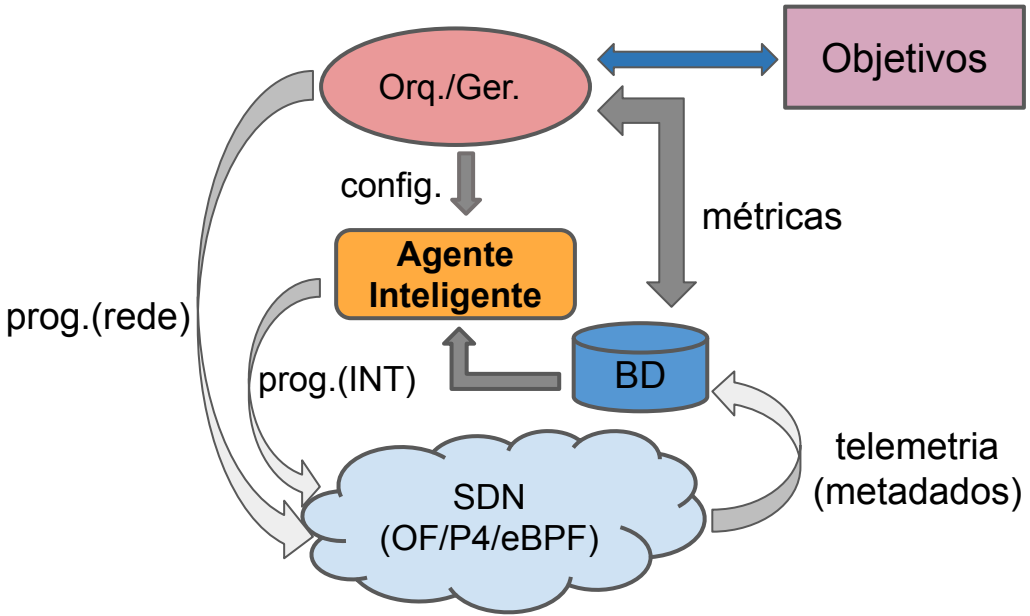
Caso tradicional:



Proposta de solução

Telemetria in-band (INT) adaptativa provida por IA

usando aprendizado por reforço profundo



+ telemetria quando necessário
- telemetria quando não for

+/- telemetria:
quantidade de metadados, nº de fluxos monitores, taxas de pacotes dos fluxos monitores, nº de SW c/ coletas, etc

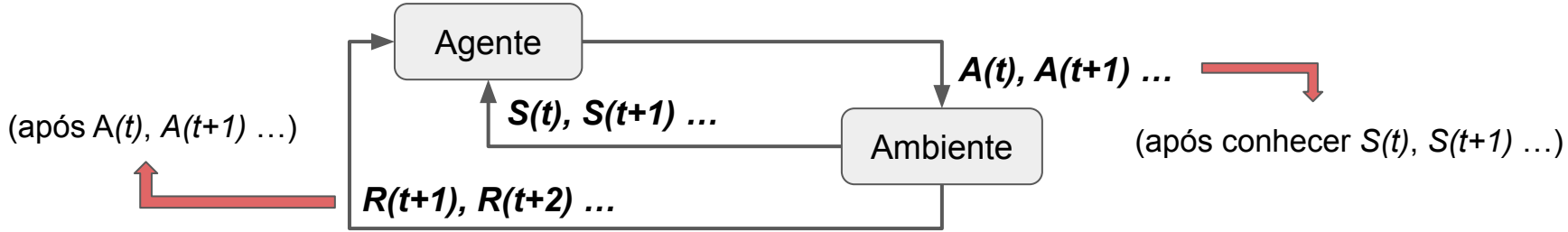
Aprendizado por reforço profundo

Adequado p/ problemas de controle em malha fechada

O agente inteligente aprende a atuar por ***tentativa e erro***, mapeando situações (**estado de um ambiente**) em **ações nesse ambiente**:

- Conforme acerta ou erra na ação tomada, o agente é recompensado positivamente ou negativamente.

O agente interage com o ambiente executando ações aleatórias durante um tempo definido (exploração), buscando aprender uma política de comportamento que maximize as recompensas acumuladas ao longo do tempo.



Aprendizado por reforço profundo

Fase de ***exploration***:

- O agente inicia o treinamento com a estratégia de ***exploration***, vivenciando o impacto de suas ações através das recompensas retornadas pelo ambiente.
- Com probabilidade definida pelo hiperparâmetro ***epsilon*** (taxa de exploração), o agente toma uma ação aleatória a cada estado recebido.

Fase de ***exploitation***:

- O agente usa o conhecimento já adquirido para tomar ações que retornem as maiores recompensas.
- A cada *timestep*, ***epsilon*** é reduzido pelo hiperparâmetro ***epsilon decay*** e, ao atingir ***epsilon min***, o agente para de "explorar" e estabiliza em ***exploitation***.

Aprendizado por reforço profundo

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

Aprendizado:

Trata-se de um **MDP** contendo uma função de probabilidade de transição de estados ***P(s, a, s')*** e uma função da recompensa imediata esperada ***R (s, a, s')***:

- ***s*** é o estado atual, ***a*** é a ação a ser executada e ***s'*** é o próximo estado.
- ***P*** fornece a probabilidade de um agente no estado ***s***, ao executar a ação ***a***, atingir o estado ***s'***.
- ***R*** retorna a recompensa esperada nesse passo.

O agente executa uma política ***π(s, a)*** que busca maximizar, a longo prazo, o valor esperado da soma das recompensas futuras:

$$V^{\pi}(s) = \mathbb{E}_{\pi, s_0=s}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots + \gamma^T r_T\right] = \mathbb{E}_{\pi, s_0=s}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$$

O hiperparâmetro $\gamma \in [0, 1]$ *penaliza as recompensas mais antigas.*

Aprendizado por reforço profundo

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

Aprendizado:

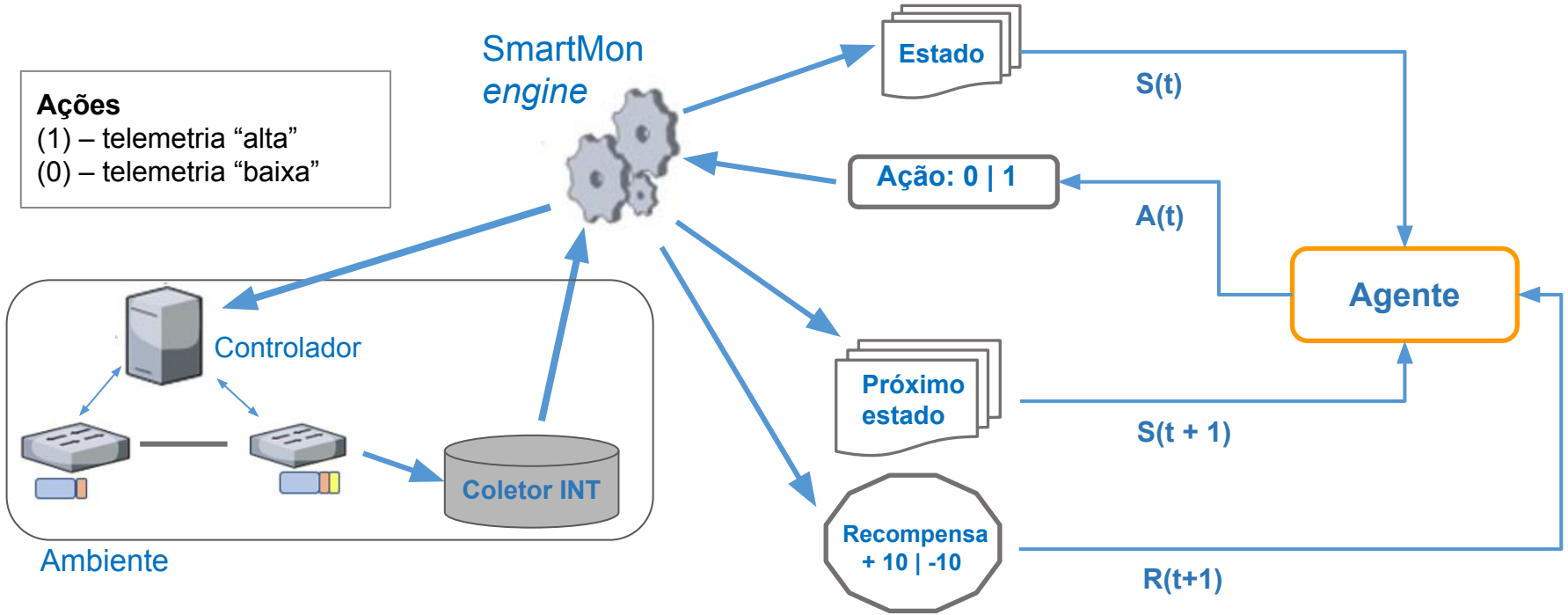
$Q^{\pi}(s, a)$ descreve a quantidade esperada de recompensas acumuladas a partir do instante em que se toma a ação ***a*** no estado ***s*** para uma política ***π***.

Permite obter a função de valor ótimo ***$Q^{*}(s, a)$*** *para todas as políticas possíveis, determinando qual ação o agente deve tomar numa dada circunstância.*

O algoritmo **Q-learning** computa ***Q(s,a)*** iterativamente, o que pode ser inviável conforme o nº de pares ***(s,a)***.

Usa-se então uma **DNN** para aproximar o valor de ***Q(s,a)***.

O *framework* SmartMon



O *framework* SmartMon

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

SmartMon:

- Cria os estados de representação do ambiente
- Orquestra as ações (manipula a telemetria)
- Executa a lógica de *"recompensação"*

Regula toda a interação entre o **agente** e o **ambiente**

Objetivo:

Telemetria ***alta*** quando houver **sobrecarga nos enlaces**
(formação de fila nos buffers de saída)

Telemetria ***alta***:
todos os metadados são inseridos ou ***maior taxa de fluxo monitor***
Telemetria ***baixa***:
[sw Ids, ingress port Ids egress port Ids, hop latencies] ou ***menor taxa de fluxo monitor***

O *framework* SmartMon

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

Representação do estado *s(t)* do ambiente

- Baseado nas métricas computadas pelo **INT Collector** e armazenadas nas tabelas do InfluxDB
- Adota uma janela de tempo de **30 seg** (empiricamente definida) para representar o estado do ambiente

Representação de ***s(t)*** contém <u>todas as tuplas de **métricas**</u> computadas pelo INT Collector ao longo de 30 seg

Métricas do INT Collector:
(diferente de 'metadados')

- ***<fluxo (5-tuple) – flow path>***
- ***<fluxo (5-tuple) – flow latency>***
- ***<fluxo (5-tuple) + sw_id – flow per hop latency>***
- ***<sw_id + queue_id – queue occupancy>***

O *framework* SmartMon

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

Lógica de "recompensação"

- Se enlace está com **sobrecarga**:
  - *ação foi de telemetria **alta**, recebe recompensa +10*
  - *ação foi de telemetria **baixa**, recebe recompensa -10*
- Se enlace está com **folga**:
  - *ação foi de telemetria **baixa**, recebe recompensa +10*
  - *ação foi de telemetria **alta**, recebe recompensa -10*

Limites empiricamente definidos p/ verificação de sobrecarga:

Hop latency = 1000
Flow latency = 1000
Queue occupancy = 1

O *framework* SmartMon

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

Rede neural profunda do agente

- Baseada na implementação de código aberto do algoritmo Deep Q-learning para cenário de jogos:

  https://github.com/keon/deep-q-learning/blob/master/dqn.py

- Incluindo os algoritmos DQN, Double DQN e Sarsa

A implementação do *framework* **SmartMon** pode ser acessada em

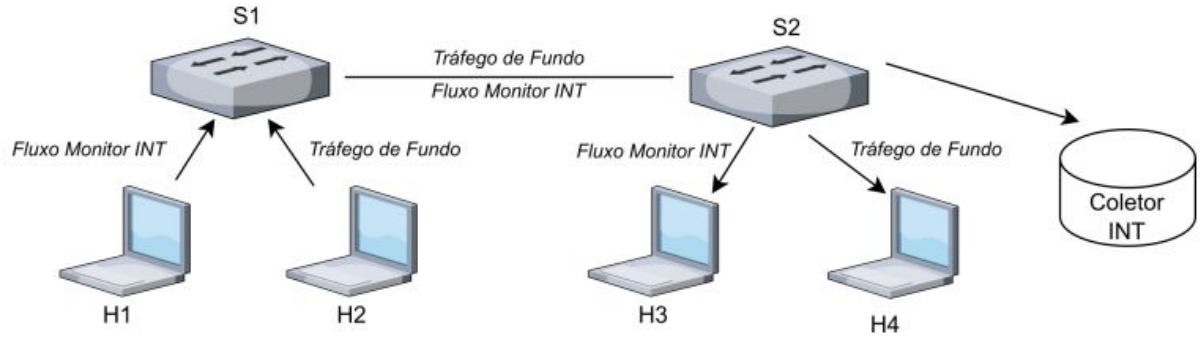
https://github.com/unirio/SmartMon

Ambiente experimental

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

Para fins de prova de conceito

- Ambiente simples e controlado p/ validar a implementação do SmartMon
- Comprovar sua capacidade de aprendizado para adaptação da telemetria
- Estudar eficiência do aprendizado para diferentes valores de hiperparâmetros



Enlaces de 50 Mbps

Mininet
ONOS c/ INT
SWs bmv2
INT Collector
InfluxDB

Taxa do fluxo monitor: 3 Mbps, empiricamente ajustada para observabilidade
Taxa do tráfego de fundo: aleatória [5, 10, 20, 25, 30, 35, **45**, **51**] Mbps

Gerado c/ **iperf**:
UDP, MTU de 1500 bytes

Ambiente experimental

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

Workflow dos experimentos

1) Tráfego de fundo alterna sua banda aleatoriamente a cada 60s ao longo do experimento
2) Injeta-se 30s de fluxo monitor para produzir as métricas de telemetria armazenadas no Influxdb
3) **SmartMon** lê essas métricas, computa ***s(t)*** e passa essa representação do estado do ambiente p/ o Agente
4) Agente define a ***"intensidade"*** telemetria que deve ser usada, com base em seu aprendizado, e a informa ao **SmartMon**
5) **SmartMon** avalia o ambiente para computar ***R(t+1)***, passa a recompensa para o Agente e aplica a telemetria na rede, voltando ao passo (2)

O tempo discreto ***(t)*** envolve o *loop* entre os passos (2) e (5).

Ambiente experimental

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

Hiperparâmetros

- **Batch size**: nº de exemplos a cada rodada de treino (o agente só aprende após acumular um *batch size* de experimentos)
- **Episódios**: conjunto de experiências (*timesteps*) p/ treino do modelo, aplicando o desconto *gamma* (**γ**)
- **epsilon**, **epsilon decay** e **epsilon mín**: regulam a transição entre as fases de *exploration* e *exploitation*
- **Rede neural profunda** (DNN) usada pelo agente

Valores usados nos experimentos:

- **epsilon** inicial = 1; **epsilon decay** = 0,97 e 0,985; **epsilon mín** = 0,01; **γ** = 0,95
- **batch size** = 16 e 32; **episódios** de 32 e 64 *timesteps*
- **DNN** c/ cam. de entrada c/ 210 (tam. dos estados); 1ª cam. c/ 128 neur., ativ. ReLU; 2ª c/ 64 neur., ativ. ReLU; e saída c/ 2 neur. (ações), ativ. linear; otimizador Adam; função de custo foi o MSE

Ambiente experimental

Métricas de avaliação:

- **PAL** e **PAD**: respectivamente, são as probabilidades de acertar **“ligar”** e de acertar **“desligar”** a telemetria alta

  ligar *– ação de ativar a telemetria alta*

  desligar *– ação de ativar a telemetria baixa*

  Serve para verificar a fração de vezes em que o **agente** tomou uma decisão correta e foi recompensado positivamente.

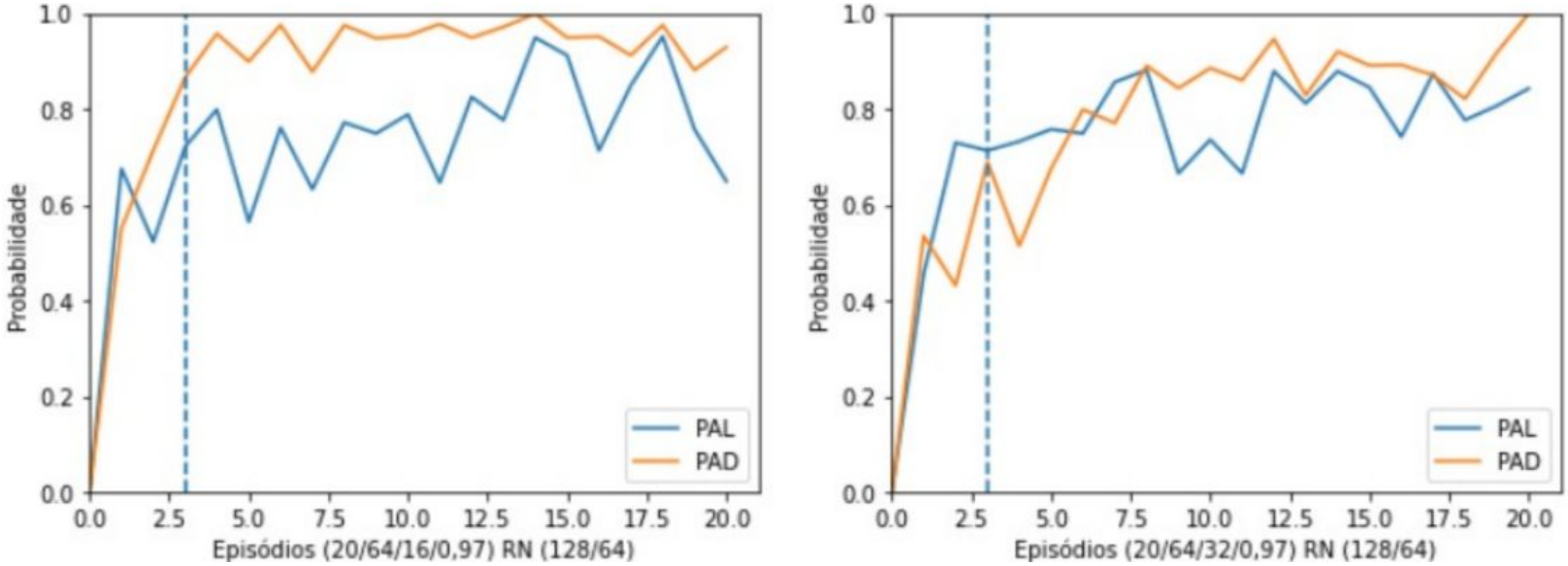
- **Recompensas acumuladas ao longo do tempo**: acumula todas as recompensas recebidas pelo agente durante os episódios de treinamento

  Verifica-se o ponto de mín. da curva; ponto de cruzamento ascendente do nível zero (caso ocorra); e o crescimento da curva após o algoritmo estabilizar.

Resultados (ação: **mais ou menos metadados** , fluxo monitor em **3 Mbps**)

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

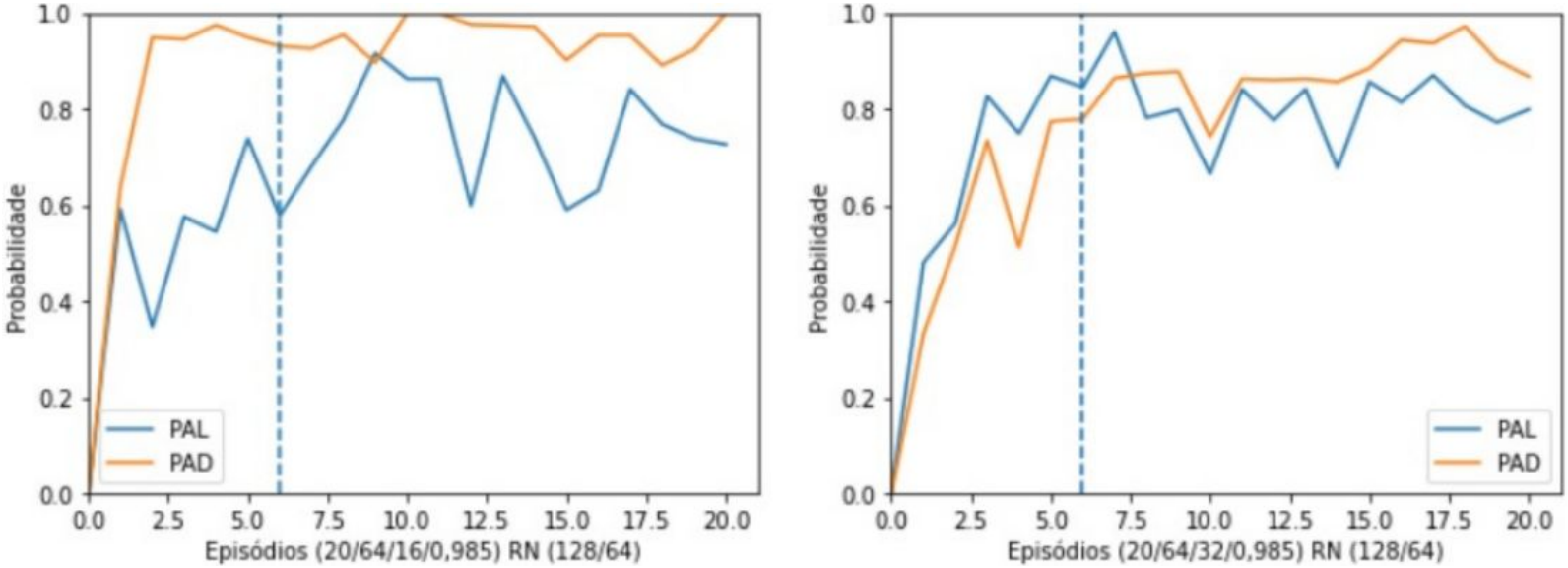
SmartMon: **episódios** de 64 *timesteps*, **batch size** variando, **epsilon decay** 0,97



*Linha tracejada vertical indica mudança da fase de **exploration** para **exploitation**.*

Resultados (ação: **mais ou menos metadados** , fluxo monitor em **3 Mbps**)

SmartMon: **episódios** de 64 *timesteps*, **batch size** variando, **epsilon decay** 0,985

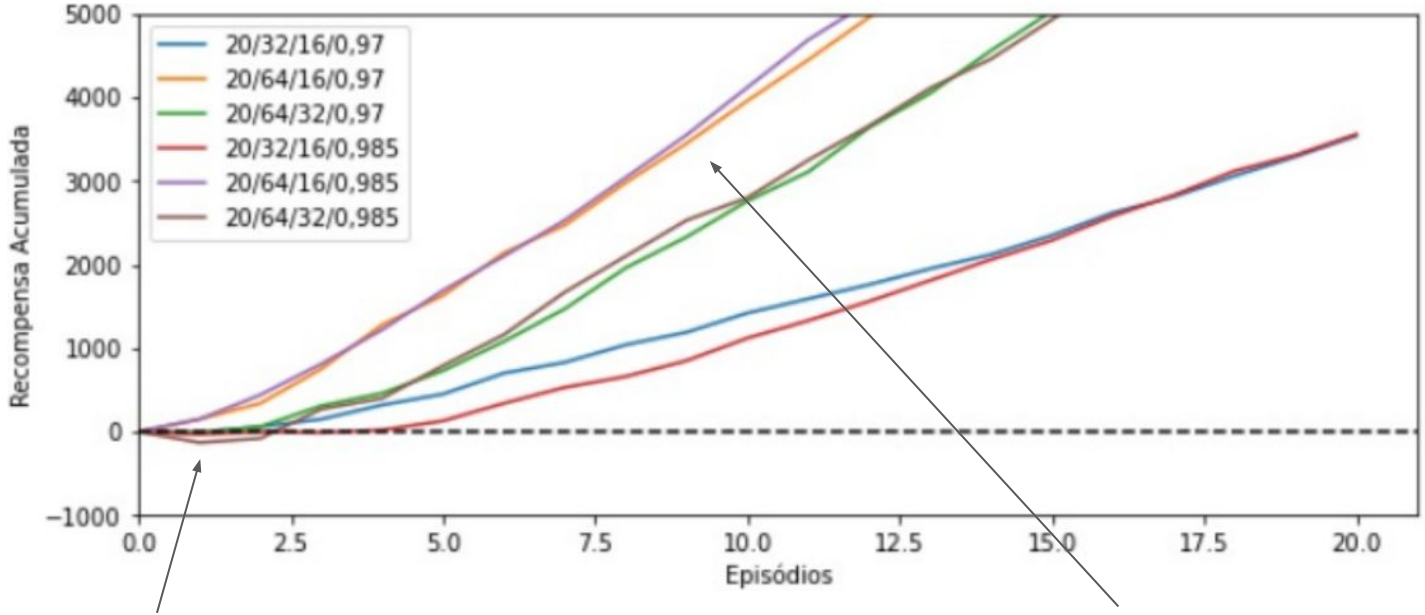


*Linha tracejada vertical indica mudança da fase de **exploration** para **exploitation**.*

Resultados (ação: **mais ou menos metadados** , fluxo monitor em **3 Mbps**)

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

SmartMon: comparativo variando hiperparâmetros

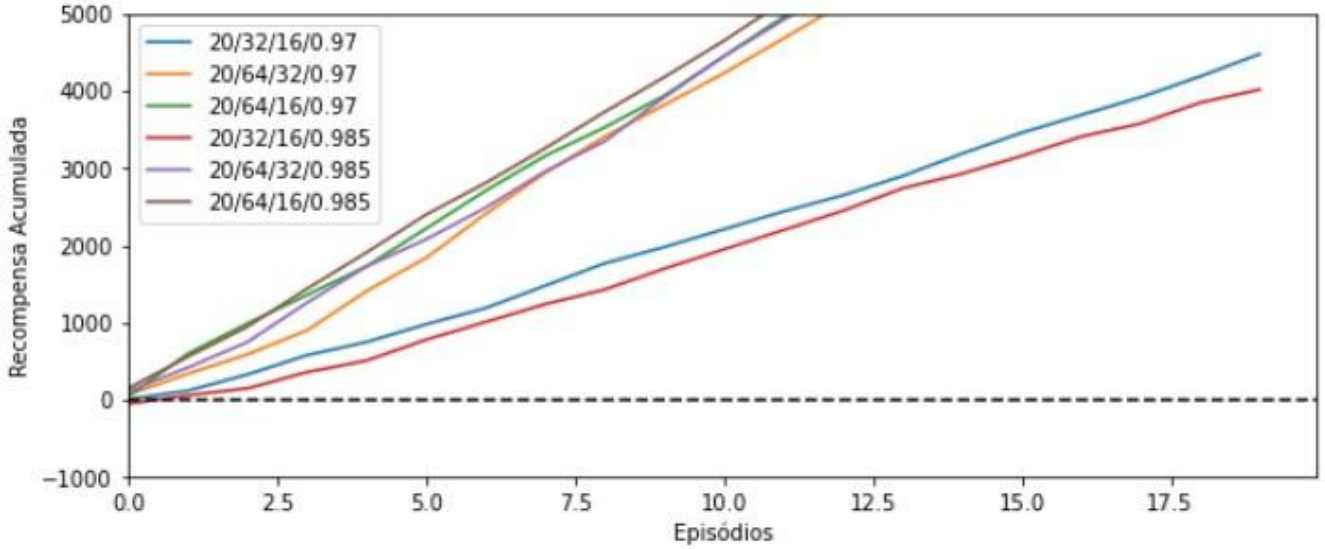


*Curva **marrom**: maior tempo explorando o ambiente. Curvas **roxa** e **laranja**: maior eficiência no aprendizado (**episódios de 64 timesteps e batch size de 16**).*

Resultados (ação: **mais ou menos metadados** , fluxo monitor em **3 Mbps**)

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

SmartMon: comparativo usando DQN de 3 camadas (128/64/32) no agente

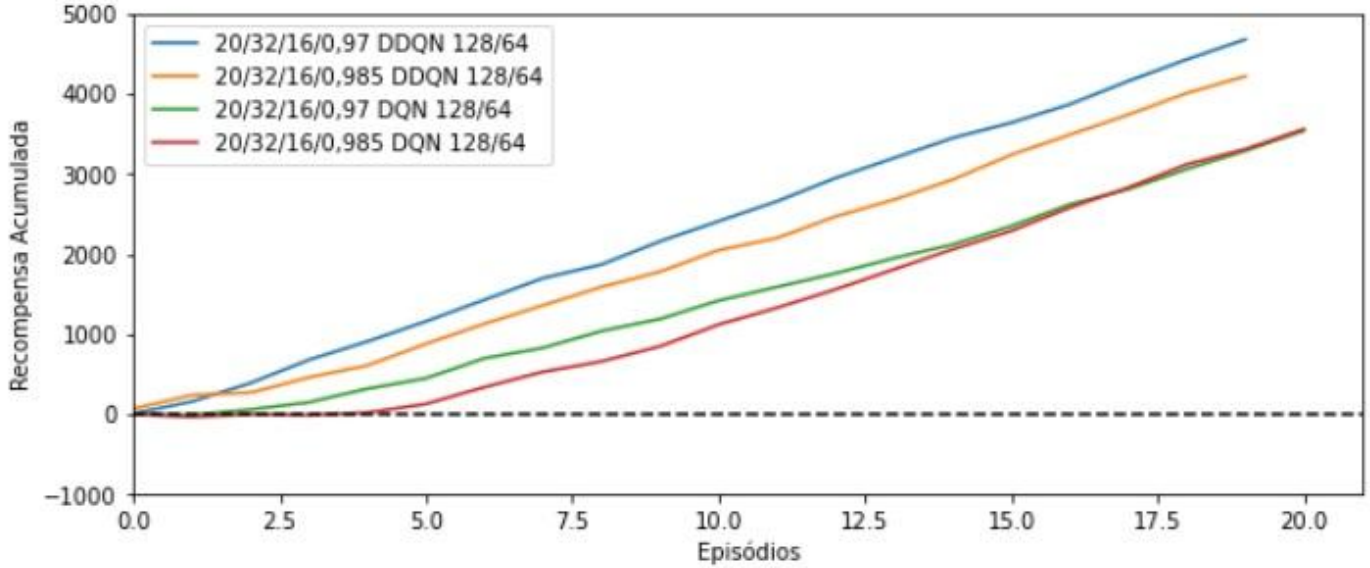


Combinações com 64 timesteps sempre têm melhor desempenho em relação às de 32 timesteps por haver o dobro de repetições por episódio para o agente aprender.

Resultados (ação: **mais ou menos metadados** , fluxo monitor em **3 Mbps**)

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

SmartMon: comparativo entre algoritmos DQN (128/64) e Double DQN (128/64)

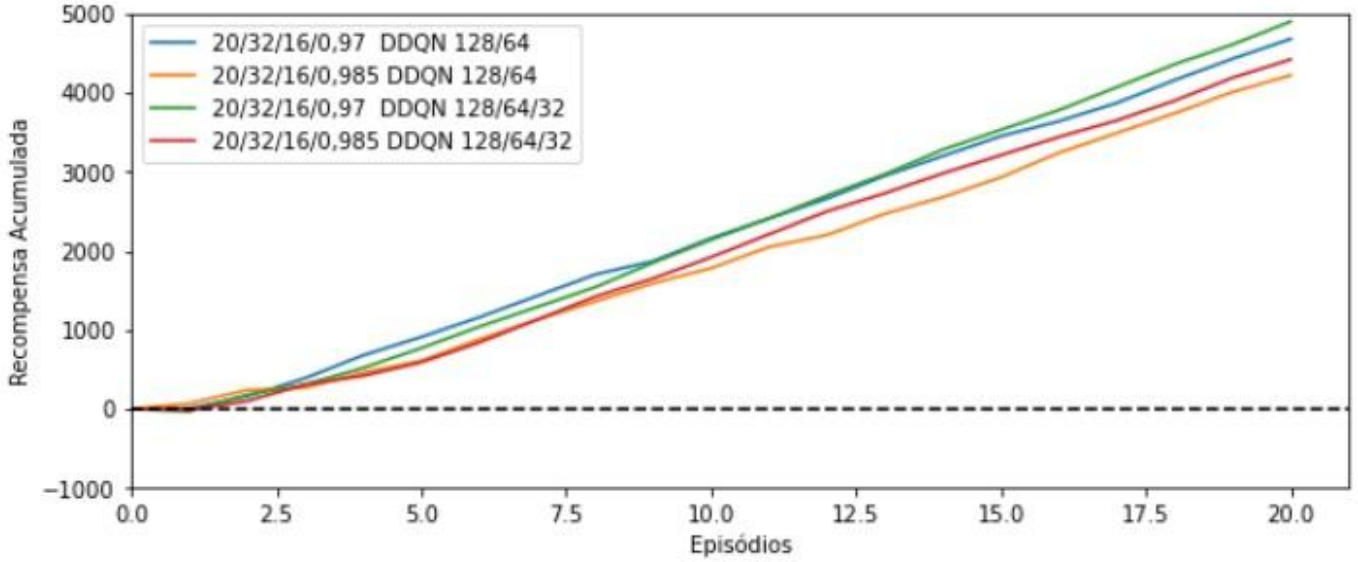


*Curvas **azul** e **laranja**: maior eficiência no aprendizado usando Double DQN no agente (episódios de 32 timesteps e batch size de 16).*

Resultados (ação: **mais ou menos metadados** , fluxo monitor em **3 Mbps**)

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

SmartMon: comparativo usando Double DQN com 2 e 3 camadas

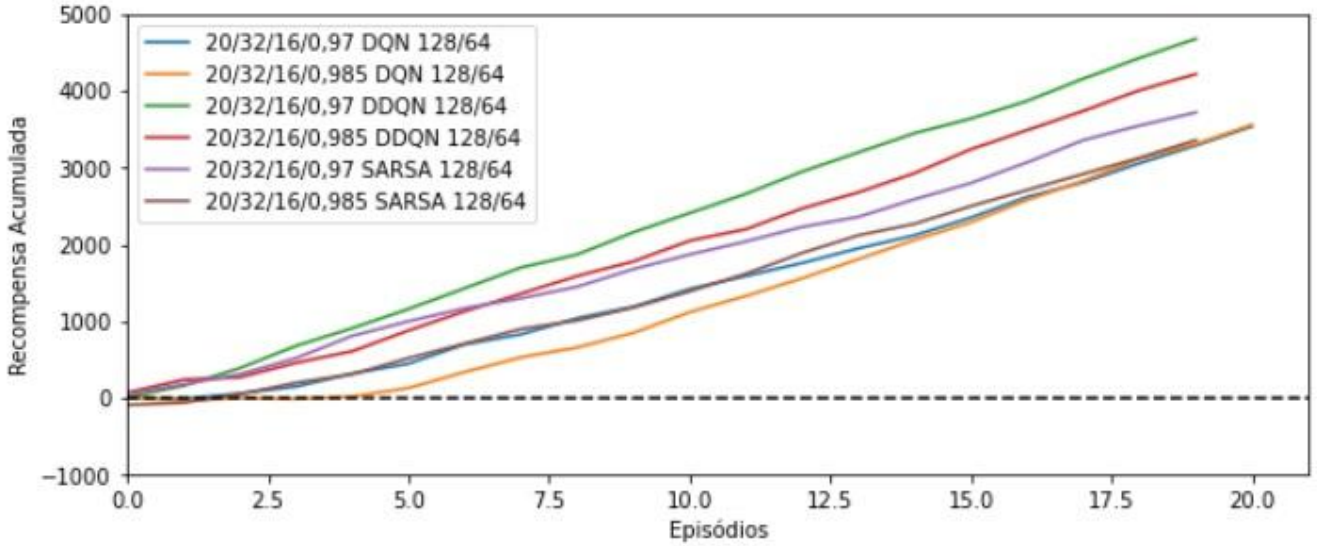


Diferença marginal entre as curvas sugere equivalência de desempenho para ambos os casos (episódios de 32 timesteps e batch size de 16).

Resultados (ação: **mais ou menos metadados** , fluxo monitor em **3 Mbps**)

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

SmartMon: comparativo entre DQN, DDQN e SARSA com 2 camadas (128/64)

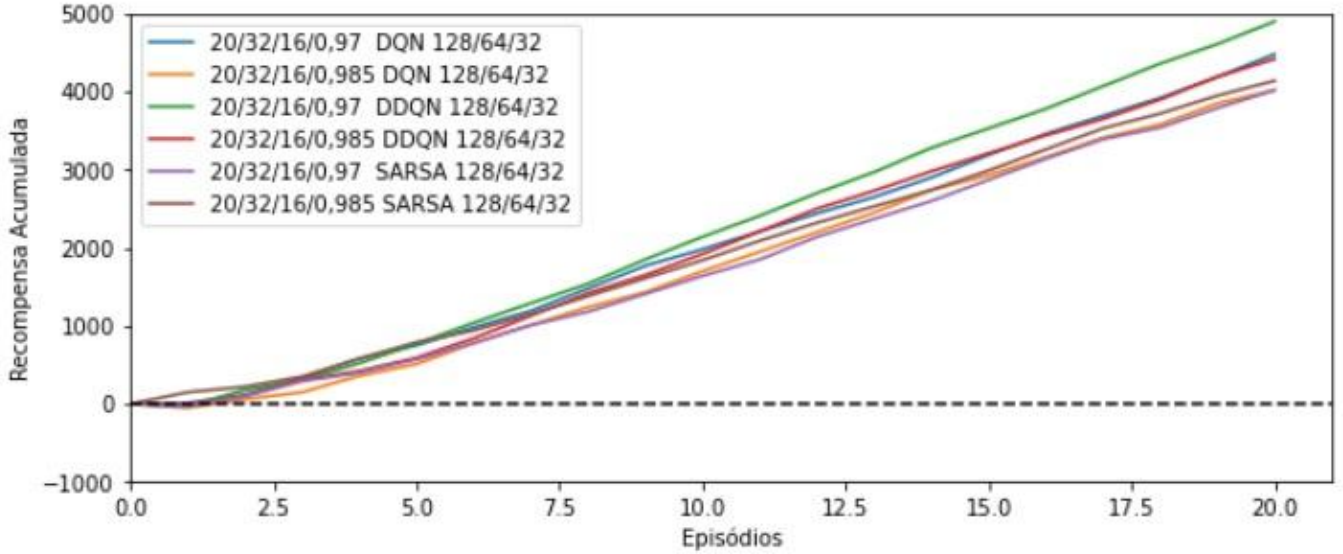


Double DQN aparenta ter um melhor desempenho (episódios de 32 timesteps e batch size de 16).

Resultados (ação: **mais ou menos metadados** , fluxo monitor em **3 Mbps**)

UNIRIO
Universidade Federal do Estado do Rio de Janeiro

SmartMon: comparativo entre DQN, DDQN e SARSA com 3 camadas (128/64/32)



Com redes neurais profundas de 3 camadas, os algoritmos DDQN, DQN e SARSA se equivalem em desempenho (episódios de 32 timesteps e batch size de 16).

Conclusão e trabalhos futuros

SmartMon: *framework* de telemetria adaptativa capaz de antever situações de congestionamento e de folga para ajustar a intensidade do monitoramento

- *Usa aprendizado por reforço profundo para aprender o comportamento na rede e decidir quando aumentar ou diminuir a telemetria*
- *Se vale de programabilidade no plano de dados para adaptar a telemetria*

Análises experimentais comprovaram a eficácia do **SmartMon** e permitiram avaliá-lo para diferentes configurações de hiperparâmetros.

Trabalhos futuros:

- Implementar no framework SmartMon um controle de intensidade de telemetria baseado na taxa de pacotes dos fluxos monitores
- Avaliar o *framework* SmartMon em cenários de rede maiores

Agradecimentos

Agradecemos a inestimável ajuda e o suporte da **Amlight-ExP** *(Americas-Africa Lightpaths Express and Protect project)* e da *Florida International University's Center for Internet Augmented Research and Assessment* (**CIARA**) pelo fornecimento do ambiente necessário para a realização das atividades experimentais.

Em especial:

- ***Jerônimo Bezerra*** e ***Ítalo da Silva Brito***

Contatos

Debora Helena Job: debora.job@uniriotec.br

Sidney Cunha de Lucena: sidney@uniriotec.br

Pedro Nuno de Souza Moura: pedro.moura@uniriotec.br

Dúvidas?