

Avaliação de Desempenho de uma Arquitetura de Sistema de Monitoramento de Pacientes em Hospitais Inteligentes com Fontes de Dados Interna e Externa

Francisco M. S. Rocha¹, Lucas Vinícius S. dos Santos¹, Francisco Airton Silva¹

¹Laboratório de Pesquisas Aplicadas a Sistemas Distribuídos (PASID)
Universidade Federal do Piauí (UFPI) - PI - Brazil

{fmarcosrocha, vinicius.lucas, faps}@ufpi.edu.br

Abstract. *The variety of sensors for health monitoring helps in decision making by medical professionals. For certain clinical situations, it is interesting to monitor the patient's health after discharge. Wearable health monitoring devices (such as smartwatches) can be used to keep sending data to the hospital. However, for monitoring a large number of patients (external and internal), a resilient and high-performance computing infrastructure is required. Such characteristics require high monetary cost equipment. To help plan such an infrastructure, this paper presents an SPN (Stochastic Petri Network) model for evaluating the performance of a multi-tier hospital system architecture (edge-fog-cloud). The model allows evaluating the mean response time (MRT), level of resource utilization (U) and probability of data loss (DP). A very specific characteristic is to consider two data sources (internal and external). As a case study, the impact of varying the number of containers (from 5 to 100) for simultaneous processing in the cloud was analyzed. With capacity 50, 75 and 100 simultaneous containers, the result was similar; with MRT below 25 seconds. For capacity 5 and 25 containers, MRT has reached 150 seconds.*

Resumo. *A variedade de tipos de sensores para monitoramento de saúde ajuda na tomada de decisão por profissionais da área. Para determinados quadros clínicos é interessante o monitoramento da saúde do paciente pós alta. Dispositivos vestíveis de monitoramento de saúde (como smartwatches) podem ser utilizados para enviar dados para o hospital. No entanto, para o monitoramento de um grande número de pacientes (externos e internos), é necessária uma infraestrutura computacional resiliente e de alto desempenho. Tais características exigem equipamentos de alto custo monetário. Para ajudar a planejar tal infraestrutura, este artigo apresenta um modelo SPN (Stochastic Petri Network) para avaliação de desempenho de uma arquitetura de sistema hospitalar multicamadas (edge-fog-cloud). O modelo permite avaliar o tempo médio de resposta (MRT), nível de utilização de recursos (U) e probabilidade de perda de dados (DP). Uma característica bem específica é considerar duas fontes de dados (interna e externa). Como estudo de caso foi analisado o impacto da variação do número de contêineres (de 5 à 100) para processamento simultâneo na nuvem. O resultado mostrou-se semelhante com capacidades 50, 75 e 100 contêineres, com MRT abaixo de 25 segundos. Para a capacidade 5 e 25 contêineres, o MRT passou de 150 segundos.*

1. Introdução

A Internet das Coisas (IoT) é um dos grandes avanços da tecnologia moderna. Os dispositivos IoT são intercomunicáveis e podem extrair informações sobre diferentes ambientes. Os dispositivos IoT são usados, por exemplo, para monitorar pacientes em hospitais inteligentes [Uslu et al. 2020], para monitorar casas inteligentes [Huang et al. 2020] ou para detectar acidentes em redes ad-hoc veiculares [Kaur et al. 2020]. O grupo *Statista*¹ prevê que os dispositivos IoT instalados devem chegar a 21,5 bilhões de unidades em todo o mundo até 2025. A IoT desempenha um papel importante no cenário da saúde. Um ótimo exemplo é como a IoT auxiliou no combate ao coronavírus em todo o mundo por meio do monitoramento de sinais vitais [Ndiaye et al. 2020, Singh et al. 2020]. A incorporação da IoT na área da saúde é denominada IoHT (Internet of Healthcare Things) [Yassein et al. 2019].

IoHT é um ecossistema de dispositivos médicos, como dispositivos inteligentes de monitoramento e assistência médica (por exemplo, marca-passo inteligente, medidor de glicose no sangue, etc.) que têm a capacidade de coletar, analisar e transmitir dados vitais dos pacientes para sistemas de saúde por meio de redes de comunicação. A computação em nuvem pode ser usada para armazenar e analisar dados de saúde, permitindo a tomada de decisão automatizada em relação às intervenções. No entanto, recursos de computação mais próximos são necessários para reduzir atrasos de processamento. *Fog e edge computing* são camadas complementares que preenchem essa lacuna de processamento local. Essa arquitetura *Cloud-Fog-Edge* fornece o suporte de processamento necessário para os dispositivos restritos de IoHT. Dessa forma, as tecnologias IoHT podem contribuir para melhorar o desempenho, reduzir custos e mitigar riscos [Rajasekaran et al. 2019].

A rede de um hospital inteligente pode receber dados de várias fontes (salas de hospital), processar dados de forma descentralizada com recursos de computação digital superiores para tomar decisões mais rápidas e assertivas. No entanto, os requisitos de sistema de hospitais inteligentes são altamente críticos, incluindo desempenho. A avaliação de desempenho de sistemas hospitalares inteligentes é essencial para garantir seu funcionamento ideal. Avaliar o desempenho de IoHT suportado por recursos *Cloud-Fog-Edge* com experimentos reais pode ser altamente caro e impraticável com pacientes reais. Existem muitos parâmetros envolvidos e modelos analíticos podem ser úteis neste contexto, fazendo previsões baseadas em probabilidades.

Existem trabalhos bem avançados que avaliam o desempenho de sistemas IoHT com uso de redes de Petri estocásticas (SPN) [Rodrigues et al. 2021, Andrade and Nogueira 2020]. O trabalho de [Nguyen et al. 2021] apresenta em seu artigo mecanismos que possibilitam o rejuvenescimento dos componentes, visto que com o tempo eles tendem a ficar mais propícios a falhas. Silva et al. [Silva et al. 2021] utilizou um modelo de filas para avaliar o desempenho através de métricas tradicionais como MRT, utilização, vazão e taxa de perdas. Os trabalhos citados acima não possuem as mesmas características do presente estudo, principalmente em relação ao aspecto de fonte de dados interna e externa ao hospital e às métricas de avaliação de desempenho.

Este artigo apresenta um modelo de redes de Petri estocásticas (SPN) para avaliar um sistema IoHT de um hospital inteligente. SPNs são modelos analíticos que

¹*Statista*: <https://tinyurl.com/y83kkser>

representam sistemas complexos com diversas características, incluindo paralelismo e concorrência. Os SPNs possuem fundamentos probabilísticos e com resultados muito confiáveis se bem aplicados. O modelo proposto considera quatro camadas de processamento (IoT, *fog*, *edge* e *cloud*) e duas fontes de dados (interna e externa). Em resumo, as principais contribuições deste artigo são:

1. Um modelo SPN para avaliar o desempenho de arquiteturas computacionais em hospital inteligente, com capacidade para configurar até 32 parâmetros (incluindo tempos de processamento e *delay*, tempos de falha e recuperação e capacidades paralelas de processamento ou armazenamento). O modelo avalia o uso de recursos computacionais, o tempo médio de resposta (MRT) e taxa de perda de dados de uma arquitetura IoHT.
2. Uma análise de sensibilidade sobre o modelo que permite identificar os fatores de maior impacto nas métricas de interesse. A análise permitiu observar, por exemplo, que as aplicações banco de dados e JBoss tiveram um grande impacto sobre a disponibilidade do sistema e, portanto potencial impacto também no desempenho.
3. Um estudos de caso usando o modelo SPN proposto que mostra como administradores de sistemas podem planejar suas infraestruturas hospitalares específicas, em especial num contexto em que é necessário mais de uma fonte de dados ao mesmo tempo.

O restante do artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados; a Seção 3 apresenta um resumo da ideia da arquitetura modelada; a Seção 4 apresenta o modelo proposto e as métricas utilizadas na avaliação; a Seção 5 mostra os resultados de um estudo de caso e por fim, a Seção 6 apresenta a conclusão do trabalho.

2. Trabalhos Relacionados

Nesta seção são abordados alguns trabalhos dentro do contexto de *smart hospitals* e *healthcare*. A literatura contempla uma variedade de estudos analíticos sobre arquiteturas que apoiam ou mantêm sistemas de *smart hospital*, no entanto relativamente poucos focam na integração do sistema interno do hospital com sensores e monitores externos.

O trabalho de [Silva et al. 2021] contextualizou uma arquitetura de smart hospital utilizando componentes IoHT e analisou as métricas MRT, DP, utilização e disponibilidade, já o trabalho de [Nguyen et al. 2021] abordou um contexto genérico de computação de borda com utilização de dispositivos móveis. Os trabalhos de [Chen et al. 2011], [Gomez-Sacristan et al. 2015], [Khayal and Farid 2021] e [Rodrigues et al. 2021] abordaram especificamente o contexto de sistemas hospitalares inteligentes e analisaram diversas métricas e comportamentos do sistema através da integração de sensores internos de monitoramento dos pacientes em conjuntos com ferramentas de auxílio de tomada de decisões por médicos de plantão.

Todos os trabalhos acima relacionados possuem em sua proposta de arquitetura apenas uma fonte de dados de requisições, já o presente trabalho bem como o trabalho de [Araujo et al. 2014] e [Andrade and Nogueira 2020] baseiam-se em uma arquitetura de smart hospital com duas entradas de dados de forma simultânea. Este último propõe uma solução de recuperação da disponibilidade do sistema por desastres ocorridos na infraestrutura do hospital, e presente trabalho foca em gerar *insights* de planejamento na

implementação da disposição e dimensionamento da capacidade dos componentes que compõe a arquitetura proposta. A característica que destaca-se neste trabalho e que não contempla em nenhum dos demais é a utilização de manipulação de filas no modelo. Nossa proposta não abstrai este aspecto, mas sim especifica capacidades limitadas de filas de entrada em cada camada da arquitetura.

Tabela 1. Comparativo dos trabalhos relacionados

Trabalho	I	II	III	IV	V
[Nguyen et al. 2021]	Computação de borda através de dispositivos móveis	MRT, utilização, DP e disponibilidade	sim	única	não
[Silva et al. 2021]	Arquitetura IoHT em smart hospital	MRT, utilização, mensagens do sistema, vazão e DP	não	única	não
[Chen et al. 2011]	Smart hospital	Quantidade de atendimentos	não	única	não
[Gomez-Sacristan et al. 2015]	Smart hospital	Delay de tráfego e largura de banda utilizada	não	única	não
[Khayal and Farid 2021]	Smart hospital	Custo por paciente e nível de recuperação de saúde	não	única	não
[Rodrigues et al. 2021]	Smart hospital	MRT, DP, número de descartes e utilização	sim	única	não
[Andrade and Nogueira 2020]	Smart hospital	Probabilidade de entrega, <i>Recovery Time Objective</i> e <i>Recovery Point Objective</i>	sim	duas	não
[Araujo et al. 2014]	Smart hospital	Probabilidade de entrega	não	duas	não
Este trabalho	Smart hospital	MRT, utilização e DP	sim	duas	sim

I - Contexto; II - Métricas; III - Análise de sensibilidade; IV - Fonte de dados; V - Manipulação de filas.

3. Arquitetura Computacional de um Smart Hospital

A Figura 1 apresenta a arquitetura modelada neste artigo. A arquitetura proposta tem o objetivo de sintetizar a interação dos componentes necessários para um sistema de *smart hospital*. A arquitetura contempla uma camada de geração de dados e três camadas em sequência de processamento dos dados. Na camada de geração de dados é composta por duas fontes de dados. A primeira é uma fonte interna que representa os sensores e dispositivos IoT que fornecem os dados do paciente dentro do hospital. A segunda fonte é uma fonte externa que representa os dispositivos que possam fornecer dados de saúde de pacientes fora do hospital pós-alta. Fora do hospital podem ser usados smartwatches e smartphones para transmitir os dados por uma rede 5G, por exemplo. De fora do hospital, os dados são transmitidos por torres de transmissão, mas esta questão pode ser abstraída caso os dispositivos tenham acesso à rede de ambientes fechados. No hospital, cada grupo de pacientes internos é designado a um roteador específico, organizado de acordo com os andares do prédio, por exemplo. As duas fontes de dados alimentam o *gateway* de borda.

Após a admissão, a requisição passa por três camadas de processamento em sequência: *edge*, *fog* e *cloud*. As camadas podem ter responsabilidades com níveis diferentes de criticidade, a *cloud* pode efetuar trabalhos mais pesados do que a *edge*, por exemplo. A camada de *fog* baseia-se em processamento paralelo em contêineres, onde cada instância conta com três aplicações: JVM, JBoss e Banco de Dados Mysql. A camada de *cloud*, bem como na *fog*, utiliza contêineres de processamento em seus dados recebidos e efetua o processamento final. O tempo de serviço para processamento de cada requisição na camada de *cloud* foi nomeada como LPT - *Last Processing Time*.

Uma questão muito importante é a abstração do que significa capacidade de processamento em cada camada. Na camada da *edge*, a capacidade refere-se ao processamento paralelo onde cada core da máquina é responsável por uma requisição. Na camada

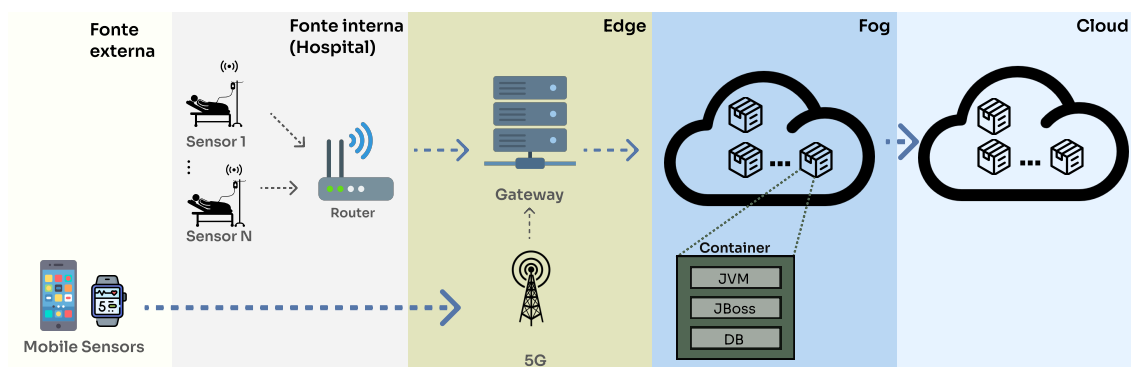


Figura 1. Arquitetura de um sistema de *smart hospital* com duas fontes de dados

de *fog* e *cloud*, a capacidade refere-se ao número de contêineres paralelos para processamento. Estes elementos de processamento podem ser abstraídos no modelo. Na SPN, tais capacidades são configuradas associando número de *tokens* (ler referências sobre SPN na Seção 4) disponíveis com respectivo tempo médio de serviço.

4. Modelo SPN

Essa seção apresenta o modelo SPN proposto (ver Figura 2). Todos os componentes do modelo seguem os mesmos padrões de nomes usados na descrição da arquitetura. A avaliação do modelo foi feita por meio da ferramenta Mercury [Maciel et al. 2017]. A dinâmica da modelagem é baseada no transporte de *tokens* entre os diferentes lugares. Para as transições entre os lugares são atribuídos tempos médios. É indicado as seguintes referências externas para conceitos aprofundados de SPN: [Rodrigues et al. 2021] [Santos et al. 2021].

O modelo é descrito a seguir da esquerda para a direita. Os *tokens* são gerados seguindo uma distribuição específica, normalmente exponencial para este tipo de contexto. Os tempos entre chegadas de requisições são setados nas transições AD1 e AD2, correspondendo às entradas interna e externa respectivamente. A antena 5G recebe as requisições externas. Os *tokens* das duas fontes convergem para a camada da *edge* e entram numa fila de espera de processamento representado pelo triângulo cujo lugar de capacidade é dado por F.E. O triângulo posterior representa a capacidade simultânea de processamento. Em seguida, os *tokens* seguem para a camada de *fog* e *cloud*. Ambas possuem filas de entrada de dados e processamento paralelo, aqui idealizadas como contêineres.

O fluxo dos *tokens* depende, além do processamento nas camadas (ver transições na Tabela 2), da disponibilidade dos componentes (Tabela 3, elaborada com os valores de [Costa et al. 2016], [Brito et al. 2021], [Santos et al. 2022]) e modelados com arcos inibidores.

Os componentes que compõem cada instância de processamento nas camadas de *fog* e *cloud* possuem, assim como os demais, restrições de disponibilidade. Para que um *token* seja processado é necessário que pelo menos um contêiner na sua camada esteja disponível, e para que o contêiner esteja disponível é necessário que suas aplicações (JVM, JBoss e DB) também estejam. O modelo tracejado em vermelho (Figura 2) é utilizado para computar os valores de MTTF_CO e MTTR_CO que representam o tempo médio de

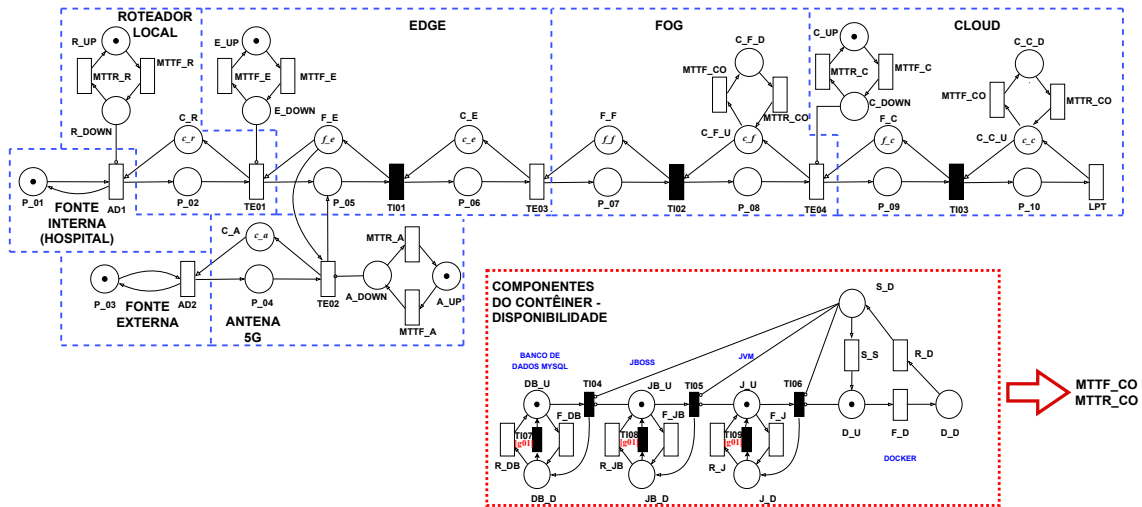


Figura 2. Modelo base proposto em SPN

Tabela 2. Descrição das principais transições do modelo base

Tipo	Elemento	Descrição	Valor (ms)
Transições temporizadas	AD1	Tempo de espera entre as requisições internas	10 à 100
	AD2	Tempo de espera entre as requisições externas	10 à 100
	TE01	Tempo de envio da requisição do router até a <i>edge</i>	20
	TE02	Tempo de envio da requisição da antena até a <i>edge</i>	20
	TE03	Tempo de processamento na <i>edge</i>	30
	TE04	Tempo de processamento na <i>fog</i>	50
	LPT	Tempo de processamento na <i>cloud</i> - último processamento/transição temporizada	250
Transições imediatas	S.S	Tempo para iniciar os serviços após a recuperação do container	500
	TI01	Encaminhamento da fila na <i>edge</i>	-
	TI02	Encaminhamento da fila na <i>fog</i>	-
	TI03	Encaminhamento da fila na <i>cloud</i>	-
	TI04, TI05 e TI06	Transições de dependência entre os componentes	-
	TI07, TI08 e TI09	Usadas pra forçar reinício dos componentes pela [g01]	-

falha e recuperação do contêiner, respectivamente.

Este trabalho realizou uma análise de sensibilidade sobre o modelo proposto [Silva et al. 2022]. A Tabela 3 apresenta um ranqueamento do índice de importância dos fatores do modelo em relação às transições relacionadas a valores de MTTF e MTTR. A transição JB_F (tempo médio para falha do JBoss) obteve o maior índice (0.7700), portanto o componente JBoss que pertence a cada instância de contêiner Docker nas camadas de *fog* e *cloud* é considerado o componente mais influente.

O modelo proposto é generalizável para outros contextos, porém foi inspirado em uma arquitetura tradicional de *smart hospital* [Rodrigues et al. 2021]. Os parâmetros de capacidades foram definidos de forma empírica. Bem como os parâmetros de transição temporizada, os quais foram setados com variação de probabilidade exponencial e influ-

Tabela 3. MTTR e MTTF dos componentes e análise de sensibilidade

Transição	Descrição	Valor (horas)	Índice de sensibilidade
F_JB	MTTF do JBoss	4728	0.7700
F_DB	MTTF do Banco de Dados	8643	0.7124
R_DB	MTTR do Banco de Dados	108	0.4521
R_D	MTTR da instância Contêiner Docker	0.255	0.3841
MTTF_R	MTTF do router	44957	0.3397
MTTF_E	MTTF da máquina de <i>edge</i>	8760	0.3128
MTTF_A	MTTF da antena	150000	0.2972
MTTF_C	MTTF da máquina de <i>cloud</i>	768.35	0.2859
R_J	MTTR da JVM	72	0.2478
F_D	MTTF da instância Contêiner Docker	2516	0.2156
F_J	MTTF da JVM	4728	0.1919
MTTR_R	MTTR do router	5	0.1749
R_JB	MTTR do JBoss	72	0.1088
MTTR_A	MTTR da antena	5	0.1001
MTTR_E	MTTR da máquina de <i>edge</i>	8	0.0461
MTTR_C	MTTR da máquina de <i>cloud</i>	0.74	0.0250

enciaram diretamente na calibração. Tais parâmetros devem ser definidos para a realidade específica do avaliador.

4.1. Métricas

As métricas adotadas nesse trabalho são: *MRT* (*Mean Response Time*), utilização individual de cada componente, e *DP* (*drop probability*). Para cálculo do *MRT* foi adotada a Lei de Little [Little 1961]. Tal lei afirma que o tempo médio de resposta (*MRT*) é proporcional ao número médio de requisições em progresso (*RequestInProgress*) e inversamente proporcional à taxa de chegada (*ArrivalRate*). A taxa de chegada é calculada como o inverso do tempo de espera entre as requisições (*ArrivalDelay*). A Lei de Little requer um sistema estável, ou seja, que possua uma taxa de chegada menor do que a taxa de processamento interno do sistema, assim ele não ficará sobrecarregado.

Normalmente o *MRT* é igual ao produto da quantidade de requisições em progresso, multiplicado pelo tempo entre chegadas de requisições, conforme *eq01* da tabela 4. No entanto, o modelo proposto possui o diferencial de ter duas entradas de dados ao mesmo tempo. Assim, para calcular o *MRT* no modelo proposto foi necessário utilizar o tempo da última transição do modelo, denominado *LPT* (*Last Processing Time*), conforme *eq02*. Para encontrar o número de *RequestsInProgress*, é necessário obter a quantidade momentânea total de *tokens* que estão sendo processados desde o início do modelo até o final. Para esse cálculo basta somar a esperança estatística de existir *tokens* em todos os lugares onde a requisição irá passar conforme *eq03*.

A métrica de Utilização para cada componente é calculada pela relação entre quantidade de *tokens* que estão à espera do processamento dividido pela capacidade máxima de seus processadores. A utilização da camada da *edge*, por exemplo, é calculada conforme equação *eq04*. A utilização da *fog* e da *cloud* seguem a mesma ideia. Já para a métrica *Drop Probability* (*DP*), são utilizadas as equações *eq05* e *eq06*. Nelas são calculadas a

probabilidade de haver perdas de requisições causadas pelo enfileiramento e consequentemente o gargalo em um ou mais componentes no momento em que a requisição chega após a admissão no sistema. Do ponto de vista prático, a requisição é descartada quando no momento em que ela tenta entrar no sistema não há mais vagas na capacidade da fila de espera. Como foram consideradas duas fontes de dados, para calcular o DP foi utilizada uma decisão de dois caminhos (*OR*) entre os DPs nas duas fontes de dados (interna e externa), conforme *eq07*.

Tabela 4. Equações utilizadas nas métricas

Ref	Equação
<i>Eq01</i>	$MRT = RequestsInProgress \times ArrivalDelay$
<i>Eq02</i>	$MRT = RequestsInProgress \times LPT$
<i>Eq03</i>	$RequestsInProgress = (Esp(\#P_06) + Esp(\#P_07) + Esp(\#P_08) + Esp(\#P_09) + Esp(\#P_10))$
<i>Eq04</i>	$U_EDGE = Esp(\#P_06)/C_E$
<i>Eq05</i>	$DP1 = (\#P_01 > 0)AND(\#C_R = 0)$
<i>Eq06</i>	$DP2 = (\#P_03 > 0)AND(\#C_A = 0)$
<i>Eq07</i>	$DP_TOTAL = P\{((\#P_01 > 0)AND(\#C_R = 0))OR((\#P_03 > 0)AND(\#C_A = 0))\}$

5. Estudo de Caso

Esta seção apresenta os resultados de desempenho do modelo proposto. A Tabela 5 apresenta os dados da parametrização do modelo com a marcação dos três macro-componentes de processamento (camada de *edge*, camada de *fog* e camada de *cloud*). A fila para redirecionamento das requisições do roteador e da antena também são definidas. É importante frisar que esses valores foram definidos no processo de calibração do modelo a fim de se obter dados mais espaçados provendo maior impacto em relação à variação utilizada e consequentemente maior riqueza nas informações. Neste estudo de caso variamos a taxa de chegada de requisições em função da variação da capacidade de processamento da *cloud*. A taxa de chegada foi variada entre 0.01 msg/ms até 0.1 msg/ms, com 20 pontos amostrais. As capacidades adotadas referem-se à quantidade de contêineres paralelos, a saber: C1 = 5, C2 = 25, C3 = 50, C4 = 75, C5 = 100. As métricas analisadas foram: MRT, utilização da *edge*, utilização da torre, utilização da *fog*, utilização da *cloud* e *drop probability* (DP).

Tabela 5. Definição da quantidade de tokens

Marcação	Descrição	Quantidade
<i>c_r</i>	Fila do roteador	50
<i>f_e</i>	Fila da <i>edge</i>	100
<i>c_e</i>	Número de canais de transmissão na <i>edge</i>	100
<i>f_f</i>	Fila da <i>fog</i>	150
<i>c_f</i>	Número de contêineres na <i>fog</i>	50
<i>f_c</i>	Fila da <i>cloud</i>	300
<i>c_c</i>	Número de contêineres na <i>cloud</i>	5, 25, 50, 75 e 100
<i>c_a</i>	Fila da antena	50

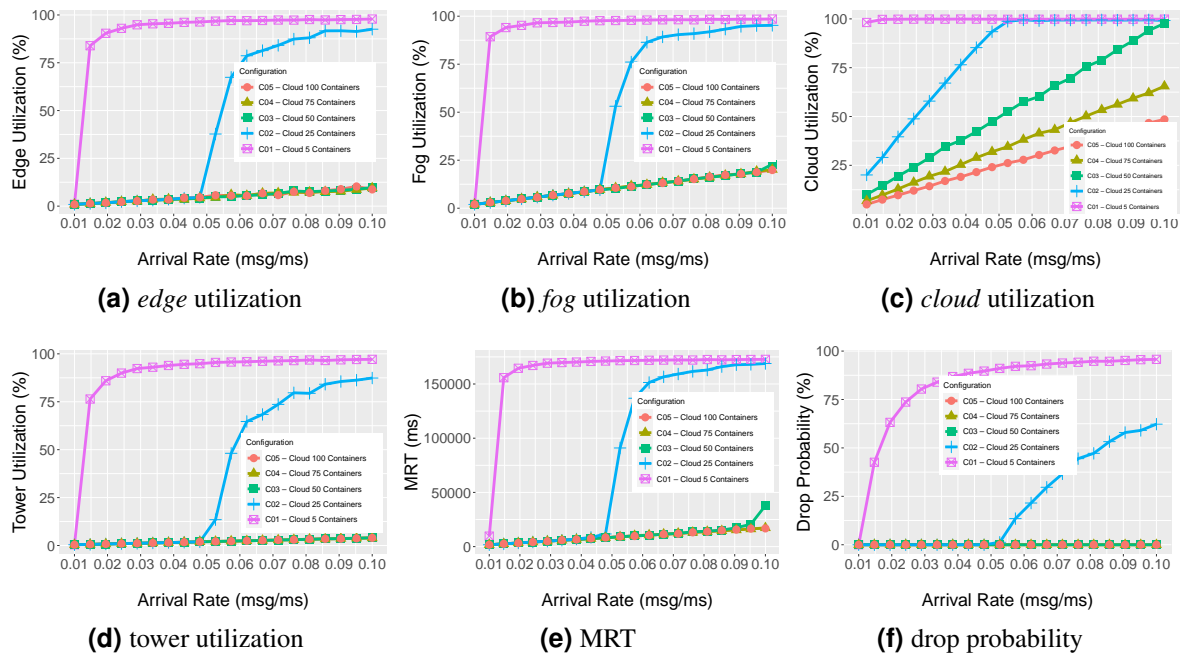


Figura 3. Análise de Desempenho do Cenário

5.1. Resultados

A Figura 3 representa os dados de desempenho referentes ao cenário analisado. A Figura 3a apresenta os dados da utilização da camada de *edge*. As configurações de 50, 75 e 100 contêineres não resultaram em alta utilização, os dados mantiveram-se sobrepostos e lineares e não ultrapassaram os 10%. Já as configurações de 5 e 25 contêineres obtiveram um alcance próximo a 100%. Um aumento acentuado logo nas menores taxas de chegada pode ser observado com a capacidade C01, saltando de menos de 1% para 83.90%, até chegar em 97.86%. Com a capacidade C02 a utilização mostrou-se significativa a partir de uma taxa média de 0.055 msg/ms, ultrapassando os 50% e chegando até 92.55% na maior taxa de chegada (0.1 msg/ms). A variação da utilização da *edge* pela taxa de chegada alcançou números elevados com essas configurações de 5 e 25 contêineres.

A Figura 3b apresenta a utilização da *fog*. Os resultados mostraram-se semelhantes aos da *edge* nas configurações de 5 e 25 contêineres. Nas configurações C3, C4 e C5 conseguiu-se atingir uma baixa utilização em torno de 20% na maior taxa de chegada. O resultado da utilização da Torre apresentado na Figura 3d assemelha-se também com a da *edge*, evidenciando uma equilibrada carga de trabalho dos componentes que recebem as requisições do sistema.

O gráfico da utilização da *cloud* 3c foi o que apresentou os dados mais variados e com menos sobreposições. A configuração de 5 contêineres resultou em uma utilização próxima aos 100% durante toda a avaliação, a configuração de 25 contêineres resultou também em uma sobrecarga de utilização da *cloud*. As demais configurações mantiveram-se lineares e proporcionais à taxa de chegada, sendo que na taxa de chegada final (0.1 msg/ms), a configuração de 50 contêineres resultou em 97.66% de utilização. A configuração de 75 contêineres obteve 65.52% e a configuração de 100 contêineres 48.52%. O resultado da configuração de 50 contêineres apresentou uma utilização da

cloud praticamente linear e proporcional à taxa de chegada, atingindo quase 100% na maior taxa e iniciando com 10% na menor taxa. Já com a configuração de 25 contêineres, a utilização apresenta índices acima de 99% quando a taxa de chegada atinge na metade da escala de avaliação (em torno de 0.055 msg/ms) e segue até o final com o mesmo comportamento da configuração de 5 contêineres. A configuração de 5 contêineres é a que mais se difere das demais em relação ao índice medido na taxa de partida, a utilização obtida já inicia com um índice de 98.15%.

A Figura 3e apresenta os dados do MRT. As configurações de 50, 75 e 100 contêineres obtiveram resultados praticamente sobrepostos. Porém, na taxa de chegada final, a de 50 contêineres obteve um tempo de resposta de 37s e as configurações de 75 e 100 contêineres atingiram um MRT máximo de 17s. O tempo de resposta com a configuração de 5 contêineres atingiu números acima de 150s nas taxas iniciais, já com a configuração de 25 contêineres, a métrica se mantém na faixa de 2s a 10s até uma taxa de 0.05 msg/ms. Uma forte aceleração do crescimento do tempo de resposta é apresentada entre a faixa média de 0.05 msg/ms e 0.06 msg/ms com a configuração C2. A configuração de 25 contêineres de fato é a que entrega os dados mais equilibrados e espaçados do experimento em relação ao MRT com o cenário proposto da análise.

A Figura 3f mostra a probabilidade de descartes (DP). Somente com as configurações C1 e C2 o sistema consegue verificar algum descarte nas requisições. Nas demais configurações o índice permanece em 0%. Com 5 contêineres, foi observado um índice inicial de 42.58%, com uma taxa de 0.01473 msg/ms e aumentando gradualmente até atingir um índice de 95.72%. Com 25 contêineres, a probabilidade só parte de 0% a partir de uma taxa de chegada de 0.0526 msg/ms, e atinge 62.26% na taxa final. Na configuração de 5 contêineres e variação da taxa de chegada entre 0.01 msg/ms e 0.02 msg/ms, o DP cresce e chega a 65%. Com a configuração de 25 contêineres, o comportamento do DP é relativamente mais linear e proporcional se comparado ao obtido na C1.

Concluindo, com uma configuração de 25 contêineres na *cloud* o sistema se torna mais controlável a possíveis aumentos repentinos na taxa de chegada das requisições do que com uma configuração de 5 contêineres, e conseqüentemente mais controlável ao crescimento do número de pacientes que entram e saem do hospital. Em C2 os dados são mais variados e espaçados conforme as escalas das métricas de avaliação. Na faixa mediana da taxa de chegada (entre 0.05 msg/ms e 0.06 msg/ms) há um “gargalo” no processamento das requisições, afetando diretamente nos resultados obtidos. A sobrecarga das requisições é facilmente notada quando a utilização da *cloud* chega a próximo de 100% com uma taxa de chegada de 0.05 msg/ms. A partir dessa taxa, todas as outras métricas aumentam seus índices (inclusive o DP parte de 0%), levando a comprovar que quando a camada de *cloud* chega à utilização máxima, todos os outros componentes e métricas avaliadas são afetados o que por sua vez poderá prejudicar no aumento do fluxo de novos pacientes.

6. Conclusão

Implementação de novas arquiteturas requerem estudos adequados, visto que devemos nos ater a questões de viabilidade e mitigação de investimentos que possam ocorrer de forma desnecessária. O modelo consegue configurar até 32 parâmetros (incluindo tem-

pos de processamento e *delay*, tempos de falha e recuperação e capacidades paralelas de processamento ou armazenamento) e possibilita avaliar o nível de utilização dos recursos computacionais (U), o tempo médio de resposta (MRT) e a taxa de perda de dados (DP) de uma arquitetura IoHT. Os resultados obtidos com a análise do modelo devem ser observados como parâmetro no planejamento de dimensionamento dos componentes reais. Portanto, este trabalho pode auxiliar os administradores de sistemas na identificação do equipamento mais adequado em termos de custo e eficiência. Como trabalhos futuros, cogitamos realizar mais análises numéricas, avaliando diferentes cenários e aplicações.

Referências

- Andrade, E. and Nogueira, B. (2020). Dependability evaluation of a disaster recovery solution for iot infrastructures. *The Journal of Supercomputing*, 76(3):1828–1849.
- Araujo, J., Silva, B., Oliveira, D., and Maciel, P. (2014). Dependability evaluation of a mhealth system using a mobile cloud infrastructure. In *2014 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 1348–1353. IEEE.
- Brito, C., Rodrigues, L., Santos, B., Fé, I., Nguyen, T.-A., Min, D., Lee, J.-W., and Silva, F. A. (2021). Stochastic model driven performance and availability planning for a mobile edge computing system. *Applied Sciences*, 11(9):4088.
- Chen, X., Thomas, N., and Harrison, M. (2011). Performance evaluation of scheduling policies in a smart hospital environment. In *2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pages 585–592. IEEE.
- Costa, I., Araujo, J., Dantas, J., Campos, E., Silva, F. A., and Maciel, P. (2016). Availability evaluation and sensitivity analysis of a mobile backend-as-a-service platform. *Quality and Reliability Engineering International*, 32(7):2191–2205.
- Gomez-Sacristan, A., Rodriguez-Hernandez, M. A., and Sempere, V. (2015). Evaluation of quality of service in smart-hospital communications. *Journal of Medical Imaging and Health Informatics*, 5(8):1864–1869.
- Huang, D. Y., Apthorpe, N., Li, F., Acar, G., and Feamster, N. (2020). Iot inspector: Crowdsourcing labeled network traffic from smart home devices at scale. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2):1–21.
- Kaur, M., Malhotra, J., and Kaur, P. D. (2020). A vanet-iot based accident detection and management system for the emergency rescue services in a smart city. In *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, pages 964–968. IEEE.
- Khayal, I. S. and Farid, A. M. (2021). A dynamic system model for personalized healthcare delivery and managed individual health outcomes. *IEEE Access*, 9:138267–138282.
- Little, J. D. (1961). A proof for the queuing formula: $L = \lambda w$. *Operations research*, 9(3):383–387.
- Maciel, P., Matos, R., Silva, B., Figueiredo, J., Oliveira, D., Fé, I., Maciel, R., and Dantas, J. (2017). Mercury: Performance and dependability evaluation of systems with exponential, expolynomial, and general distributions. In *2017 IEEE 22nd Pacific Rim international symposium on dependable computing (PRDC)*, pages 50–57. IEEE.

- Ndiaye, M., Oyewobi, S. S., Abu-Mahfouz, A. M., Hancke, G. P., Kurien, A. M., and Djouani, K. (2020). Iot in the wake of covid-19: A survey on contributions, challenges and evolution. *IEEE Access*, 8:186821–186839.
- Nguyen, T. A., Fe, I., Brito, C., Kaliappan, V. K., Choi, E., Min, D., Lee, J. W., and Silva, F. A. (2021). Performability evaluation of load balancing and fail-over strategies for medical information systems with edge/fog computing using stochastic reward nets. *Sensors*, 21(18):6253.
- Rajasekaran, M., Yassine, A., Hossain, M. S., Alhamid, M. F., and Guizani, M. (2019). Autonomous monitoring in healthcare environment: Reward-based energy charging mechanism for iomt wireless sensing nodes. *Future Generation Computer Systems*, 98:565–576.
- Rodrigues, L., Gonçalves, I., Fé, I., Endo, P. T., and Silva, F. A. (2021). Performance and availability evaluation of an smart hospital architecture. *Computing*, 103(10):2401–2435.
- Santos, G. L., Gomes, D., Silva, F. A., Endo, P. T., and Lynn, T. (2022). Maximising the availability of an internet of medical things system using surrogate models and nature-inspired approaches. *International Journal of Grid and Utility Computing*, 13(2-3):291–308.
- Santos, L., Cunha, B., Fé, I., Vieira, M., and Silva, F. A. (2021). Data processing on edge and cloud: a performability evaluation and sensitivity analysis. *Journal of Network and Systems Management*, 29(3):1–24.
- Silva, F. A., Brito, C., Araújo, G., Fé, I., Tyan, M., Lee, J.-W., Nguyen, T. A., and Maciel, P. R. M. (2022). Model-driven impact quantification of energy resource redundancy and server rejuvenation on the dependability of medical sensor networks in smart hospitals. *Sensors*, 22(4):1595.
- Silva, F. A., Nguyen, T. A., Fé, I., Brito, C., Min, D., and Lee, J.-W. (2021). Performance evaluation of an internet of healthcare things for medical monitoring using m/m/c/k queuing models. *IEEE Access*, 9:55271–55283.
- Singh, R. P., Javaid, M., Haleem, A., and Suman, R. (2020). Internet of things (iot) applications to fight against covid-19 pandemic. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*.
- Uslu, B. Ç., Okay, E., and Dursun, E. (2020). Analysis of factors affecting iot-based smart hospital design. *Journal of Cloud Computing*, 9(1):1–23.
- Yassein, M. B., Hmeidi, I., Al-Harbi, M., Mrayan, L., Mardini, W., and Khamayseh, Y. (2019). Iot-based healthcare systems: a survey. In *Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems*, pages 1–9.