

Uma Política de Inserção de Conteúdo baseada na Correlação entre Medidas de Centralidade para Redes Centradas em Conteúdo

Igor Carvalho¹, Antônio Jorge Gomes Abelém¹

¹Grupo de Estudos em Redes e Comunicação Multimídia - GERCOM - UFPA

Abstract. *In-network caching is a key feature of Information Centric Networks (ICN). Choosing the nodes where content should be cached has been a great challenge for designing caching policies in ICN. A good way of doing this is by network centrality measures, which describe the importance of a node regarded to a feature. However, the large number of existing centrality measures poses a challenge to select the one for choosing the caching nodes. Moreover, it is unknown whether the selected measure will produce a high performance in different scenarios, because of its topological structure dependency. Because of that, a good way is to consider their correlation in order to select the nodes. In this work, we propose a caching policy for ICN based on the correlation of network centrality measures. This correlation is obtained through Pearson's correlation coefficient so that the strong or very strongly correlated ones are picked for selecting the caching nodes. Through simulation and using real backbone topologies, our proposal outperformed the other evaluated caching policies in terms of cache hits and average latency time.*

Resumo. *O cache em rede é uma importante característica de Redes Centradas em Conteúdo (RCCs). A escolha dos nodos que armazenarão o conteúdo é um grande desafio e uma boa maneira de fazer isso é através de medidas de centralidade de rede, que descrevem a importância de um nodo, dada uma característica. Entretanto, o grande número de medidas torna a escolha dos nodos ainda mais desafiadora, pois é incerto se uma medida escolhida resultará num alto desempenho em diferentes cenários, dada a sua forte dependência da estrutura topológica. Por conta disso, uma boa alternativa seria considerar a correlação entre elas para selecioná-los. Neste trabalho, propõe-se uma política de inserção de conteúdo baseada na correlação entre medidas de centralidade para selecionar àquelas forte ou mais fortemente correlacionadas para armazenar o conteúdo nos seus nodos correspondentes. Por meio de simulação e utilizando uma boa variedade de topologias nos testes, nossa proposta superou o desempenho das políticas de inserção em termos de taxa de acertos de conteúdo e tempo médio de download.*

1. Introdução

As Redes Centradas no Conteúdo (RCCs) permitem uma nova maneira de comunicação para a Internet. Nesta nova abordagem, a comunicação é baseada no conteúdo requisitado e não mais centrada na comunicação entre sistemas finais. Em RCCs, o conteúdo é solicitado e encaminhado pelo nome e qualquer entidade da rede (usuários, servidores de

conteúdo e roteadores) pode prover tal conteúdo. Além desta característica, tais entidades podem armazenar o conteúdo em *cache*, reduzindo o tempo de *download* e diminuindo a carga nos servidores [Laoutaris et al. 2006].

O *cache* em rede é uma importante característica de RCCs por conta da redução do número de saltos necessários para obtenção de conteúdo. Grande parte das requisições são satisfeitas em roteadores intermediários, chegando a uma redução de até 30 % no número de pacotes enviados. Quando se trata de tráfego inter-domínio, essa redução alcança 11 %, melhorando muito o desempenho tanto da rede como das aplicações [Sun et al. 2014]. Mas para que este aumento de desempenho seja possível, a política de inserção de conteúdo precisa ser eficaz para escolher os roteadores que armazenarão conteúdo.

Uma forma de se escolher tais roteadores é considerando tanto características topológicas quanto dos nodos da rede. Duas delas bastante importante são a topologia da rede e medidas de centralidade [Kutscher et al. 2014]. A topologia da rede influencia bastante no desempenho da mesma, dado que nodos estratégicos são escolhidos para armazenarem conteúdo. Já as medidas de centralidade são uma excelente maneira de selecioná-los, haja vista que os nodos mais bem ranqueados, dentro da respectiva medida, podem ser usados para *cache*. No entanto, elaborar uma política de inserção de conteúdo baseado em medidas de centralidade é um grande desafio, uma vez que existe um grande número delas e não se sabe se a medida escolhida terá um bom desempenho em todas as topologias de redes utilizadas. Um outro desafio é que, por conta da dinamicidade das requisições de conteúdos, a política de inserção baseada numa única medida pode resultar em baixo desempenho quando a rede atingir uma estabilidade nos padrões das requisições.

Este artigo propõe uma política de inserção de conteúdo para RCCs baseada na correlação de medidas de centralidade de rede. Argumenta-se que os nodos usados como *cache* devem ser selecionados de acordo como o nível de correlação entre tais medidas, não apenas baseado em uma única medida. Um alto nível de correlação sugere uma forte similaridade entre as medidas consideradas e, como consequência disso, os seus respectivos nodos produzem um desempenho semelhante ao serem usados pela política. Para obter tal nível de correlação, este trabalho utilizou o coeficiente de correlação de *Pearson*, que mede correlação entre duas variáveis lineares [Statstutor 2017]. As medidas de centralidade escolhidas foram: *Intermediação*, *Proximidade*, *Grau*, *PageRank* e *Katz*.

O restante deste artigo divide-se da seguinte forma: Seção 2 aborda os trabalhos relacionados; Seção 3 apresenta algumas definições importantes e detalha as medidas usadas; Seção 4 descreve o funcionamento da proposta; Seção 5 descreve a avaliação da proposta, parâmetros usados e outras configurações; Seção 6 comenta os resultados e faz algumas considerações e observações sobre os mesmos e a Seção 7 conclui o artigo.

2. Trabalhos Relacionados

O trabalho proposto por [Chai et al. 2012] usa a centralidade de intermediação para escolher os nodos para armazenarem o conteúdo. De acordo com o trabalho, a política de inserção de *cache* padrão em RCCs, LCE (*Leave a Copy Everywhere*), armazena uma cópia do conteúdo solicitado em todos os nodos ao longo do caminho em que foi solicitado. Para isso, eles usam a centralidade de intermediação para obter um melhor desempenho que a LCE. A proposta foi avaliada utilizando uma árvore *k-ary*, uma rede *scale-free*

e um *backbone* real. A proposta baseada na centralidade de intermediação atingiu um melhor resultado em relação à todas as métricas avaliadas, quando comparada com a LCE e com uma abordagem de inserção que escolhe os nodos aleatoriamente.

No trabalho proposto por [Guan et al. 2012], algumas medidas de centralidade foram utilizadas para escolher os nodos alvos de *cache*. Foram escolhidas as centralidades de grau, proximidade, intermediação e *eigenvector*. Para cada uma delas, escolheu-se os 20 nodos mais bem ranqueados. O objetivo era verificar qual medida obteria uma maior redução no número de saltos para a obtenção de conteúdo. Constatou-se que a centralidade de intermediação foi a que obteve um melhor desempenho e, portanto, o conteúdo será armazenado nos seus 20 nodos mais bem ranqueados. Apesar de considerarem uma variedade de medidas, os testes foram feitos em apenas uma topologia, e não se sabe se tal medida teria um bom desempenho em outras topologias. Além disso, outras métricas de desempenho não foram avaliadas, como, por exemplo, taxa de acertos de conteúdos e redução no tempo de *download*.

O trabalho de Rossi et al. [Rossi and Rossini 2012] propôs uma avaliação do desempenho de *caching* em CCN (*Content Centric Networks*) baseada em medidas de centralidade, na qual o tamanho do *cache* dos nodos é baseado em tais medidas. Eles exploram diferentes medidas como intermediação, proximidade, *stress*, grafo, excentricidade e grau a fim de atribuir tamanhos distintos de *cache* para os nodos. O artigo concluiu que o ganho trazido pelo uso de *caches* heterogêneos é bem limitado. Além disso, a centralidade de grau foi classificada como "suficientemente boa" para a alocação de conteúdo.

A proposta de [Yufei et al. 2016] usa uma política de inserção de conteúdo baseada na centralidade de intermediação como métrica para escolher os nodos que armazenarão o conteúdo no caminho de requisição. Quando um conteúdo necessita ser descartado de algum *cache*, tal conteúdo é enviado pelo nodo correspondente a um nodo vizinho. A avaliação da proposta considerou quatro topologias reais e ela foi comparada com as políticas LCE, *ProbCache* e *Betweenness* (Centralidade de Intermediação). A política proposta obteve um melhor desempenho que as demais, porém, ela pode perder em desempenho em grandes topologias devido à sobrecarga advinda da troca de mensagens de controle entre nodos vizinhos para a transferência de conteúdo.

3. Medidas de Centralidade

Esta seção explica brevemente o que são medidas de centralidades, algumas aplicabilidades e também detalha individualmente quais medidas foram escolhidas para compor a política de inserção proposta neste trabalho.

As medidas de centralidade são usadas para identificar nodos importantes de uma determinada topologia de rede baseada em suas propriedades estruturais [Grando et al. 2016]. Tais medidas apresentam uma natureza heurística e, por isso, não há uma definição formal a respeito do que elas devem medir ou mensurar. Além disso, não existe uma relação necessária entre um dado problema e uma medida de centralidade [Bell et al. 1999]. Por conta disso, algumas delas podem ser usada para resolver a mesma classe de problemas.

Desta forma, este trabalho selecionou cinco medidas de centralidade considerando a sua relevância na literatura, aplicabilidade e importância em outros contextos como

análise de redes complexas e de redes sociais [Chapela et al. 2015]. Vejamos cada uma delas.

- **Centralidade de Intermediação:** A centralidade de intermediação de um nodo v é a soma da fração de todos pares de caminhos mais curtos que passam por v e é dada pela Equação 1:

$$Bw(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (1)$$

onde σ_{st} é o número total de caminhos mais curtos que partem do nodo s para o nodo t e $\sigma_{st}(v)$ é o número daqueles que passam por v . Esta medida quantifica o número de vezes em que um nodo age como uma ponte ligando outros nodos. Quando maior o valor de tal centralidade para algum nodo, maior é o fluxo de algo que passa por eles, como, por exemplo, pacotes em uma rede. Tomando-se um grafo com m vértices e n arestas, foi proposto um algoritmo que leva um tempo computacional de $\mathcal{O}(mn)$ [Brandes 2001]

- **Centralidade de Grau:** É definida como o número de adjacências de um nodo. Também pode ser entendida como um nodo consegue pegar qualquer tipo de dado que está fluindo na rede. Vale ressaltar que o grau dos vértices, a distribuição do grau dos nodos, média do grau dos nodos e mínimo e máximo grau são outras métricas importantes relacionadas à essa medida. A centralidade de grau é dada pela Equação 2:

$$Dg(v) = \sum_{i=1}^n a(v_i, v_k) \quad (2)$$

onde $a(v_i, v_k) = 1$ if (v_i, v_k) são adjacentes, 0 caso contrário. A complexidade computacional de tal medida é $\Theta(n^2)$ para um grafo denso e $\Theta(m)$ para um esparso.

- **Centralidade de Proximidade:** Definida pela soma do inverso das distâncias geodésicas a partir de um vértice analisado para todos os outros, conforme mostra a Equação 3:

$$Cls = \frac{1}{\sum_{i=1}^n d(p_i, p_k)} \quad (3)$$

onde $d(p_i, p_k)$ é a distância geodésica do nodo p_i ao nodo p_k . Ao passo que a centralidade de grau considera vizinhos a um salto de distância, esta considera todo os nodos do grafo, ranqueando aqueles que apresentam uma distância média mais curta para todos os outros nodos. Além disso, pode-se entendê-la como o *tempo-até-a-chegada* dos pacotes fluindo numa rede [Borgatti 2005]. Assim como na de intermediação, esta medida tem um custo $\mathcal{O}(mn)$ para calcular todos valores de proximidade dos nodos.

- **Centralidade PageRank:** É uma medida famosa utilizada na máquina de busca do Google para o ranqueamento de páginas *webs*. Basicamente, ela define uma medida de centralidade para as páginas num *web*-grafo [Page et al. 1999]. Seja $A = (a_{i,j})$ a matriz de adjacência de um grafo direcionado. A centralidade *PageRank* Pg_i de um nodo i é dada pela Equação 4:

$$Pg_i = \alpha \sum_k \frac{a_{k,i}}{d_k} Pg_k + \beta \quad (4)$$

onde α e β são constantes e d_k é o grau de saída do nodo k se tal grau é positivo, ou $d_k = 1$ se o grau de saída k é nulo. Na forma matricial, tem-se a Equação 5:

$$Pg = \alpha x D^{-1} A + \beta \quad (5)$$

onde β é agora um vetor cujo os elementos são todos iguais à uma dada constantes positiva e D^{-1} é a matriz diagonal com o i -th elemento da diagonal igual a $1/d_i$. Esta medida reporta a influência de um nodo na rede. Essa importância diz respeito a estar conectado a vizinhos importantes, cujo alcance estende-se além das suas conexões diretas com outros nodos.

- **Centralidade Katz:** Considerando um nodo i , a centralidade *Katz* é dada pela Equação 6:

$$K_i = \alpha \sum_j A_{ij} K_j + \beta \quad (6)$$

onde A é a matriz de adjacência de um grafo G com autovalores λ . O parâmetro β controla o valor inicial de centralidade dos nodos e $\alpha < \frac{1}{\lambda_{max}}$. Esta medida calcula a influencia relativa de um nodo dentro de uma rede, medindo o número de vizinhos intermediários (nodos de primeiro grau) e também todos os outros nodos na rede que se ligam ao nodo em questão através dos tais nodos intermediários. A complexidade computacional para esta medida é tradicionalmente $\mathcal{O}(n^3)$, mas um algoritmo mais eficiente leva um tempo $\mathcal{O}(n + m)$ [Foster et al. 2001].

4. Descrição da Proposta

Por isso, propõe-se uma política de inserção de conteúdo que considera cinco medidas de centralidades (descritas na seção anterior) como base para escolher os nodos que armazenarão o conteúdo solicitado. Dentre essas cinco medidas, seleciona-se aquelas que possuem um nível de correlação *forte* ou *muito forte*, calculado através do Coeficiente de Correlação de *Pearson*. Ao fazer isso, o conteúdo é armazenado em nodos cujas as medidas possuem um alto grau de similaridade e, conseqüentemente, podem melhorar ainda mais o desempenho das RCCs.

4.1. Detalhes da Implementação

Nossa proposta foi implementada e testada utilizando o simulador de *caching* Icarus [Saino et al. 2014]. Descrevemos algumas de suas características importantes para um melhor entendimento do funcionamento de nossa política.

- **Controle Centralizado da Rede - CCR:** O simulador é baseado neste controle central para realizar as operações na rede. Por conta disso, uma política de inserção de conteúdo utiliza algumas das operações providas pelo CCR, as quais destacam-se o cálculo de rotas, obtenção dos valores de medidas de centralidades para uma topologia, diâmetro da rede entre outros.

- *Modelo Genérico de Arquitetura ICN*: O simulador não está atrelado a nenhuma das arquiteturas de ICN já implementadas como, por exemplo, CCN e NDN. Por padrão, todas as requisições de conteúdo são enviadas para o servidor de conteúdo através do cálculo da rota feita pelo CCR. Em cada salto na rede, o conteúdo solicitado é procurado nos *caches* dos nodos intermediários entre o cliente e o servidor. Uma vez encontrado, o CCR decide em qual nodo o conteúdo será armazenado no caminho reverso, conforme está implementado na política de inserção. Portanto, é a política quem decide onde o conteúdo será armazenado, mas é o CCR que executa isso na rede.

4.2. Seleção dos Nodos

A seleção dos nodos para armazenar o conteúdo foi dividida em duas partes: Fase *Off-line* e Fase *On-line*. Aquela faz referência ao cálculo dos valores de todas as medidas de centralidade escolhidas para os nodos de uma dada topologia para, então, obter o nível de correlação entre elas. Esses valores de centralidade e sua correlação podem ser obtidos através de qualquer ferramenta ou biblioteca disponível para tal e utilizados sempre que conveniente. A fase *On-line* inicia com o processo de requisição de conteúdo e, a partir do nível de correlação obtido na fase anterior, decide que medidas foram escolhidas e, conseqüentemente, que nodos armazenarão o conteúdo. Estes detalhes de implementação foram explicados a fim que nossa proposta pudesse ser melhor compreendida, já que em RCCs a tomada de decisão sobre armazenamento de conteúdo é distribuída.

Fase *Off-Line*: Antes de descrevermos o funcionamento desta fase, apresentaremos algumas definições importantes usadas ao longo do texto. Considere uma topologia representada pelo grafo $G = (V, E)$. A seguir estão as notações utilizadas no restante deste artigo:

- V : Conjunto de nodos da topologia, onde $V = [v_1, v_2, v_3 \dots v_{|v|}]$, em que $|v|$ representa o número de nodos.
- E : Conjunto de arestas que conectam os nodos.
- P_{req} : Caminho de requisição de conteúdo, onde $P_{req} = [v_1, v_2, v_3 \dots v_s]$ para $v \in V$ e v_s é um servidor de conteúdo.
- k : O conteúdo solicitado.
- P_{rv} : O caminho reverso de requisição de um nodo que possui o conteúdo até o cliente, onde $P_{rv} = [v_1, v_2, v_3 \dots v_r]$ para $v \in V$, v_r é o cliente que solicitou k . Este nodo que fornece k pode ser ou o servidor de conteúdo ou um nodo intermediário.
- C : Conjunto de medidas de centralidade, onde $C = [c_1, c_2, c_3 \dots c_n]$.
- $x_{v,c}$: O valor de uma medida $c \in C$ para o nodo $v \in V$.
- S : Conjunto contendo as medidas forte ou muito fortemente correlacionadas, onde $S = [s_1, s_2, s_3 \dots s_n]$.

Primeiramente, precisa-se calcular o valor das centralidades para cada nodo da topologia utilizada. O Algoritmo 1 recebe como entrada a topologia e o conjunto de centralidades utilizadas. Para cada valor $c \in C$, calcula-se o seu valor para o nodo v . Após isso, este valor é salvo no nodo para uso futuro. Depois, calcula-se a correlação para cada par de centralidades c_1 e $c_2 \in C$. Vale ressaltar que o número total de comparações feitas entre as medidas é igual a C_n^2 , em que n é o número total de medidas de centralidades, tomadas duas a duas. Além disso, o algoritmo não adiciona medidas já avaliadas e que

já estão em S . Utilizou-se o coeficiente de correlação de *Pearson* [Statstutor 2017], que calcula a correlação entre duas variáveis. O coeficiente de *Pearson* é definido conforme a Equação 8:

Algorithm 1: Calcular Valor Centralidade

Input: Topologia, C

```

foreach  $v \in Topologia$  do
  | foreach  $c \in C$  do
  | |  $x_{v,c} \leftarrow \text{CalculaValorCentr}(v, c)$ 
  | | Salvar( $x_{v,c}, v$ )
  | end
end

```

Algorithm 2: Calcular Correlação entre Centralidades

Input: Topologia, C
Output: S

```

foreach  $v \in Topologia$  do
  | foreach  $c_1 \neq c_2 \in C$  do
  | |  $Corr \leftarrow \text{ObterPearsonCorr}(x_{v,c_1}, x_{v,c_2})$ 
  | | if  $Corr \geq 0.6$  then
  | | | Salvar( $c_1, c_2, S$ )
  | | end
  | end
end

```

$$r = \frac{\sum_{i=1}^n (c1_i - \bar{c1})(c2_i - \bar{c2})}{\sqrt{\sum_{i=1}^n (c1_i - \bar{c1})^2} \sqrt{\sum_{i=1}^n (c2_i - \bar{c2})^2}} \quad (7)$$

Tabela 1. Classificação do Nível de Correlação do Coeficiente de *Pearson*

Valores	Classificação
0.0 - 0.19	Muito Fraca
0.20 - 0.39	Fraca
0.40 - 0.59	Moderada
0.60 - 0.79	Forte
0.80 - 1.0	Muito Forte

em que n é o número de amostras; $c1_i, c2_i$ são os valores de duas medidas indexadas por i e $\bar{c1} = \frac{1}{n} \sum_{i=1}^n c1_i$. O valor do coeficiente r fica entre $-1 \leq r \leq 1$, em que valores positivos denotam uma correlação positiva; valores negativos denotam uma correlação negativa; um valor igual a 0 representa nenhuma correlação entre as variáveis e quanto mais perto de 1 ou -1, maior o nível de correlação entre elas. O Algoritmo 2 descreve como encontrar tal correlação. Ele recebe como entrada a topologia, cujos os nodos já possuem os valores de centralidades a serem comparados, além do conjunto de centralidades C , selecionadas para análise. Para cada par de centralidades $c_1, c_2 \in C$, o coeficiente de *Pearson* é calculado utilizando-se os valores de centralidades do nodo ,

neste caso x_{v,c_1} , x_{v,c_2} , já computados no Algoritmo 1. Se o valor da correlação obtido for maior ou igual a 0.6, isso significa que as duas medidas comparadas são, no mínimo, fortemente correlacionadas, conforme a Tabela 1. Assim, c_1 e c_2 são salvas em S , caso ambas ou uma das duas não estejam em S .

Algorithm 3: Enviar Requisição de Conteúdo

```

Input:  $P_{req}, k$ 
foreach  $v \in P_{req}$  do
  if  $k$  armazenado em  $v$  then
    | EncontrarNodosSelecionados ()
  else
    | EnviarReqConteudo
  end
end

```

Fase *On-Line*: Esta fase começa com o processo de requisição de conteúdo na rede, já sabendo-se quais centralidades obtiveram uma classificação forte ou muito forte, obtido na fase anterior. Considere que um usuário envia uma requisição para um conteúdo k através de P_{req} . Segundo o Algoritmo 3, k é procurado em cada nodo de P_{req} . Em caso afirmativo, o procedimento *Encontrar Nodos Seleccionados* do Algoritmo 4 é invocado para encontrar os nodos que armazenarão k e, então, enviá-lo ao usuário que o solicitou. Caso contrário, a requisição do conteúdo k é enviada para os nodos restantes de P_{req} .

Algorithm 4: Encontrar Nodos Seleccionados

```

Input:  $P_{rv}, k, S$ 
foreach  $v \in P_{rv}$  do
  foreach  $s \in S$  do
    | if  $v == \text{NodoMaxValorCentr}(s, P_{rv})$  then
      | | ArmazenarConteudo ( $k, v$ )
    | end
  end
  | EnviarConteudoCliente ( $k$ )
end

```

Para cada medida forte ou muito fortemente em S , encontra-se o nodo que obteve o maior valor de centralidade em P_{rv} . Enquanto o conteúdo é enviado ao usuário, verifica-se se o nodo corrente é um dos escolhidos para que k seja armazenado. Em caso afirmativo, k é armazenado no nodo analisado. Após armazenar k em todo os nodos escolhidos, k é enviado ao cliente.

Para um entendimento melhor do funcionamento da proposta, tomemos o seguinte exemplo. Vamos assumir que $C = [\text{Intermediacao}, \text{Grau}, \text{Proximidade}, \text{PageRank}, \text{Katz}]$. Assume-se que após calcular os valores do coeficiente de *Pearson* para a topologia em questão, o conjunto $S = [\text{Intermediacao}, \text{PageRank}, \text{Katz}]$. Sendo assim, o conteúdo será armazenado em três nodos selecionados dentro do caminho reverso P_{rv} , que serão o com maior valor de medida de intermediação, o com maior valor de medida PageRank e o com maior valor da medida Katz. É importante enfatizar que se o conteúdo solicitado for encontrado num nodo entre o usuário e o servidor de conteúdo, este processo ocorrerá a partir deste

nodo fornecedor do conteúdo até o usuário solicitante. Isto faz com que o conteúdo seja armazenado em nodos mais próximos do usuário, reduzindo o tempo de *download* e o tráfego na rede.

5. Avaliação da Proposta

A política de decisão proposta foi implementada no simulador Icarus 0.6.0. O catálogo de conteúdos é de 1×10^4 e o tamanho médio dos pedaços do conteúdo (*chunk*) é de 15 Kb, ao passo que o tamanho médio do conteúdo é de 150 Kb (10 *chunks*); o número de requisições avaliadas foi de 2×10^6 e elas seguem um processo de Poisson com uma taxa de 100 requisições por segundo, considerando todos os clientes da rede; A política de troca de conteúdo usada foi a LRU (*Least Recently Used*); a popularidade dos conteúdos é modelada segundo a distribuição Zipf, tendo α variando entre 0.6 e 1.2, o que é uma boa aproximação para conteúdos web, conteúdos gerado por usuários, compartilhamento de arquivos e vídeo sob demanda (VoD) [Fricker et al. 2012]; o tamanho do *cache* dos nodos é de 1% e este valor é em relação ao número de conteúdos no catálogo, ou seja, os nodos terão até 100 entradas para armazenar conteúdo. As medidas de centralidade avaliadas foram *Intermediação*, *Grau*, *Proximidade*, *PageRank*, *Katz*, *PROBCACHE* (*Probabilistic Caching*), *LCE* e a nossa *PROPOSTA*.

5.1. Métricas Avaliadas

- *Taxa de Acertos de Conteúdo*: É a relação entre o número de acertos de conteúdo em nodos intermediários e o total de requisições enviadas.
- *Tempo de Download*: Tempo médio para a obtenção de pedaços (*chunks*) de conteúdo.
- *Diversidade de Conteúdo*: Esta métrica foi desenvolvida para calcular o número de elementos armazenados em *caches* intermediários. Ela expressa a taxa de armazenamento de conteúdos únicos pelos *caches* dos nodos. A Diversidade varia entre $[\frac{1}{|N|}, 1]$: Se a Diversidade tende para $\frac{1}{|N|}$, há muitas cópias do mesmo conteúdos pela rede; caso contrário, se a Diversidade tende a 1, os *caches* armazenam mais *chunks* distintos. É importante salientar que apenas um conteúdo requisitado foi selecionado aleatoriamente para se analisar tal métrica. Como nossa política de inserção faz o uso de várias medidas, avaliamos o impacto disso no desempenho da proposta.

Tabela 2. Valores do Coeficiente de *Pearson* obtidos

Topologia	Int-Gr	Int-Prox	Int-Kat	Int-Pag	Prox-Gr	Prox-Kat	Prox-Pag	Gr-Kat	Gr-Pag	Kat-Pag
Viatel	0.81	0.54	0.58	0.45	0.27	0.35	0.13	0.69	0.96	0.92
Ion	0.63	0.52	0.53	0.49	0.21	0.22	0.19	0.98	0.98	0.94
USCarrier	0.52	0.55	0.60	0.46	0.36	0.45	0.19	0.98	0.97	0.92
Cogentco	0.50	0.63	0.62	0.61	0.36	0.46	0.21	0.97	0.97	0.90
Tiscali	0.36	0.66	0.46	0.68	0.57	0.62	0.45	0.84	0.93	0.63
Kentucky	0.47	0.55	0.60	0.47	0.25	0.35	0.10	0.98	0.96	0.91

5.2. Topologias utilizadas

Selecionamos seis topologias do Internet Topology Zoo ¹ com diferentes características topológicas, tais como: número de nodos(ordem), diâmetro, grau médio dos nodos, den-

¹<http://www.topology-zoo.org/>

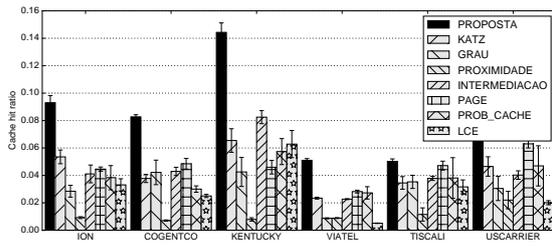


Figura 1. Taxa de Acertos - $\alpha = 0.8$

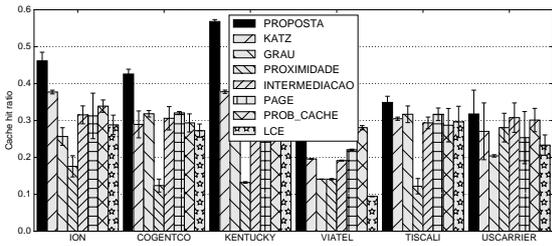


Figura 2. Taxa de Acertos - $\alpha = 1.2$

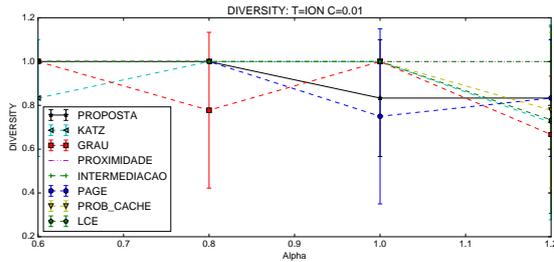


Figura 3. Diversidade - ION

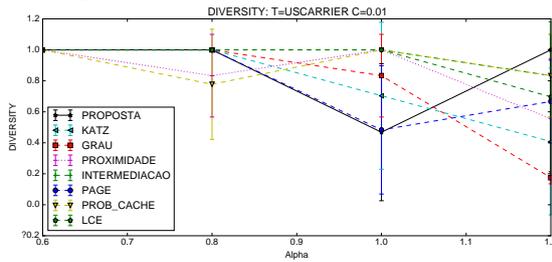


Figura 4. Diversidade - USCARRIER

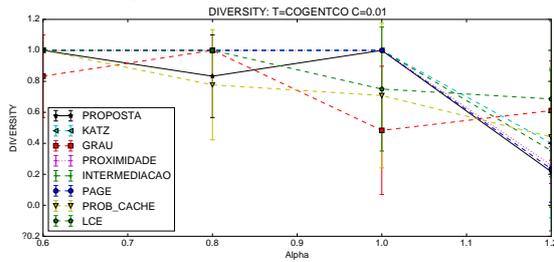


Figura 5. Diversidade - COGENTCO

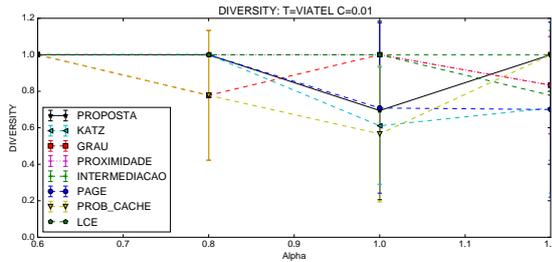


Figura 6. Diversidade - VIATEL

sidade e comprimento médio dos caminhos mais curtos. Em nossa implementação, utilizamos a biblioteca *Numpy*² para calcular o coeficiente de correlação de *Pearson* para todas as topologias usadas. A Tabela 2 contém os valores dos Coeficientes de *Pearson* obtidos; a Tabela 3 resume as informações topológicas, e a Tabela 4 contém a configuração topológica para os testes.

6. Resultados

As Figuras 1 e 2 mostram os resultados para a Taxa de Acertos para ambos valores de α . Verifica-se que a nossa proposta superou todas as outras políticas nesta métrica. Em alguns casos esse ganho foi bem acentuado, como nas topologias Ion, Cogentco e Kentucky. Já para as topologias Viatel, Tiscali e USCarrier o ganho foi um pouco próximo da segunda melhor. Como pode ser observado nos gráficos, a medida de intermediação não obteve nem o segundo melhor desempenho na maioria dos casos, mostrando que nem sempre deve ser a melhor escolha para tal. Considerando a Tabela 2, verifica-se que, dentre o conjunto de cinco medidas utilizadas, para todas as topologias avaliadas, o número de medidas selecionadas foi alto. Nas topologias Ion, USCarrier e Kentucky foram selecionadas 4 medidas. Já para as topologias Viatel, Tiscali, e Cogentco todas as

²<http://www.numpy.org/>

Tabela 3. Informações Topológicas

Topologia	Ordem	Diâmetro.	Grau médio	Densidade	Compr.Médio Cam.
Viatel	92	31	2.08	0.022	13.07
Ion	125	25	2.33	0.018	10.13
USCarrier	158	35	2.39	0.015	12.9
Cogentco	197	28	2.46	0.012	10.51
Tiscali	240	14	3.36	0.014	5.50
Kentucky	754	58	2.37	0.003	22.72

Tabela 4. Configuração Topológica

Topologia	Clientes	Roteadores	Servidores
Viatel	8	83	1
Ion	11	117	33
USCarrier	32	101	8
Cogentco	21	160	16
Tiscali	36	160	44
Kentucky	41	552	161

medidas avaliadas foram escolhidas. Mesmo com uma redundância maior deste conteúdo na rede, nossa proposta conseguiu uma taxa de acertos acima das outras propostas, pois não é apenas o fato de armazenar mais cópias de conteúdo, mas disponibilizando-os por nodos estratégicos por toda a rede, aumentando sua disponibilidade eficazmente.

Sabendo que mais nodos seriam escolhidos para armazenamento, verificamos o impacto disso na diversidade de conteúdos. Para isso, selecionamos 4 topologias para avaliar a métrica Diversidade de conteúdo. Escolhemos 2 topologias que selecionaram 4 medidas (Ion e USCarrier) e outras duas que selecionaram todas as 5 medidas avaliadas (Cogentco e Viatel). Conforme pode ser observado nas Figuras 3 - 6, nossa proposta gerou um aumento na redundância de conteúdos como esperado, considerando os valores de alfa avaliados. O pior resultado foi para a topologia Cogentco, para $\alpha = 1.2$. Quanto maior for o valor deste parâmetro na distribuição Zipf, maior a quantidade de conteúdo com popularidade similares armazenados nos *caches* dos nodos. Todas as outras políticas avaliadas também tiveram um aumento na redundância, para alguns valores de alfa. Isso mostra que os nodos intermediários não conseguiram fornecer o conteúdo solicitado e, por isso, a requisição alcançou o servidor de conteúdo, fazendo com que mais cópias fossem, também, espalhadas pela rede. Mesmo tendo aumentando o número de nodos selecionados, a nossa proposta conseguiu se sair melhor que as demais políticas para valores de alfa a partir de 1.0 nas topologias Viatel, Ion e USCarrier. Por um lado, uma diminuição na diversidade de conteúdos na rede faz com que outros conteúdos que venham a ser populares possam ser recuperados dos *caches*, o que, dependendo da dinamicidade da rede e das requisições, pode resultar em um baixo desempenho da política de inserção. Por outro lado, a nossa proposta tem um diferencial de não escolher quaisquer nodos, mas aqueles que contemplam características diferentes que, agregadas, promovem um melhor desempenho na entrega de conteúdos.

Com relação ao tempo de *download* dos conteúdos, o desempenho foi bem melhor que as demais políticas, como pode ser observado nas Figuras 7-12. Nossa proposta obteve um menor tempo para quase todas as topologias, exceto para Tiscali. Nesta topologia, a proposta obteve um tempo semelhante que a medida *Katz*. Analisando as informações topológicas da Tabela 3, a topologia Tiscali tem um comprimento médio de caminhos

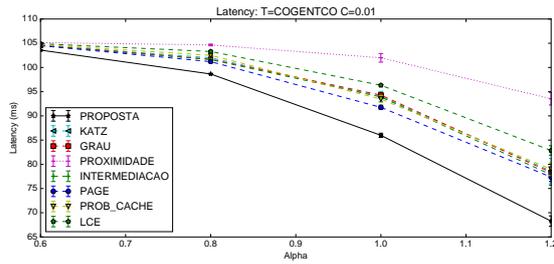


Figura 7. Latência - COGENTCO

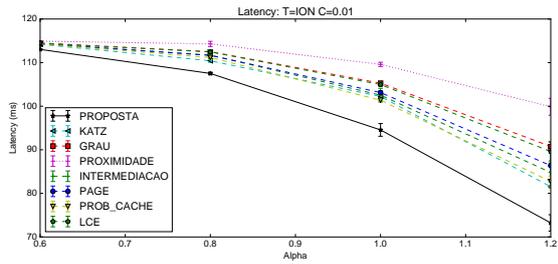


Figura 8. Latências - ION

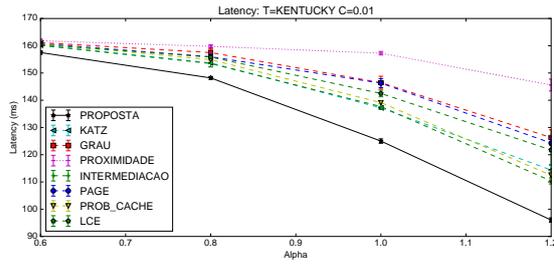


Figura 9. Latência - KENTUCKY

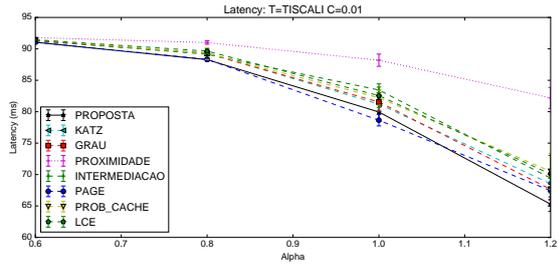


Figura 10. Latência - TISCALI

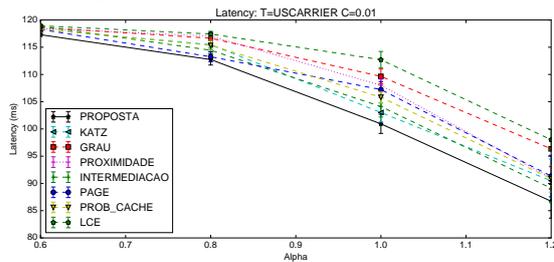


Figura 11. Latência - USCARRIER

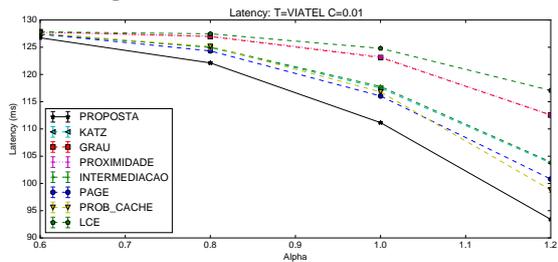


Figura 12. Latência - VIATEL

mais curtos relativamente menor que as demais topologias. Isso fez com que não houvesse uma redução considerável no número de saltos para a obtenção de conteúdo, sendo assim o desempenho obtido pela nossa proposta similar à medida *Katz*. Já para todas as outras topologias que possuem uma média de comprimento maior houve uma redução considerável no número de saltos, resultando num desempenho melhor para esta métrica. Vale destacar que até para as política *ProbCache* e *LCE*, com um funcionamento totalmente diferente das políticas baseadas em centralidade, a nossa proposta foi bem melhor que elas também.

Apesar de nossa proposta se valer de um modelo centralizado para a tomada de decisão sobre quais nodos armazenarão os conteúdos, tal decisão em ICN é distribuída. Uma possibilidade de nossa proposta ser utilizada em cenários distribuídos é o uso de algoritmos distribuídos escaláveis para o cálculo de valores de centralidades. Um trabalho recente propôs uma forma de calcular medidas de centralidade em redes complexas [You et al. 2017]. Primeiramente, propuseram um algoritmo distribuído com um tempo de convergência finito para o cálculo das medidas de proximidade e intermediação. Segundo, eles reformularam o algoritmo que calcula a medida *PageRank* e proveram um algoritmo Kaczmar-Markoviano com um tempo de convergência bem rigoroso para calcular tal medida, mesmo se o tamanho da rede é desconhecido.

A partir dos resultados obtidos, concluiu-se que:

- Medidas de Centralidade constituem-se uma ótima alternativa para a seleção de nodos estratégicos para armazenamento de conteúdo em ICN, dada a sua natureza heurística de resolverem a mesma classe de problemas, neste caso, armazenamento de conteúdo em nodos estratégicos em RCCs. Além disso, nossa proposta permite uma maior flexibilidade no desenvolvimento de políticas de inserção de conteúdo baseado em medidas de centralidades, pelo fato de cinco delas serem consideradas e não apenas uma.
- Medidas de Centralidade promoveram um ganho no desempenho quando comparado com modelos de armazenamento probabilístico (ProbCache) para topologias maiores. Esses ganhos foram similares ou menores para topologias menores ou que possuem um comprimento médio dos caminhos mais curtos menor. Porém, ao se trabalhar com medidas fortemente correlacionadas este ganho no desempenho é maior do que a abordagem probabilística.
- A redução na diversidade de conteúdo era esperada em nossa proposta, já que mais nodos são escolhidos para *cache*. Mesmo assim, a taxa de acerto manteve-se maior que as demais propostas, aumentando a permanência do conteúdo na rede e evitando que tais requisições chegassem ao servidor de conteúdo. Com isso, o tempo de *download* também foi reduzindo, otimizando a entrega do conteúdo aos usuários.

7. Conclusão e Trabalhos Futuros

Este artigo propôs uma política de inserção de conteúdo baseado na correlação entre medidas de centralidades para RCCs. A partir dos testes efetuados, comprovou-se uma melhora considerável no desempenho das métricas avaliadas, quando comparadas às políticas baseadas em um única medida, bem como à outras duas políticas com funcionamento diferente. Mesmo selecionando um número maior de nodos estratégicos para armazenar conteúdo, nossa proposta obteve um desempenho razoavelmente bom com relação à diversidade de conteúdo na rede. Algumas características topológicas como, por exemplo, um tamanho menor do comprimento médio dos caminhos mais curtos podem influenciar no desempenho da métrica tempo de *download*, como observado na avaliação. Futuramente, pretendemos selecionar, dentro das medidas forte e muito fortemente selecionadas, quais delas seriam as mais apropriadas para uma dada topologia, a fim de melhorar ainda mais os resultados obtidos. Além disso, pretendemos implementar nossa proposta numa arquitetura de RCCs como, por exemplo, NDN (*Named Data Networking*). Sabe-se que para isso algumas modificações precisam ser implementadas tanto no pacote de interesses quando nos nodos propriamente ditos para que a proposta funcione num ambiente distribuído, característico de RCCs.

Referências

- Bell, D. C., Atkinson, J. S., and Carlson, J. W. (1999). Centrality measures for disease transmission networks. *Social networks*, 21(1):1–21.
- Borgatti, S. P. (2005). Centrality and network flow. *Social networks*, 27(1):55–71.
- Brandes, U. (2001). A faster algorithm for betweenness centrality. *Journal of mathematical sociology*, 25(2):163–177.

- Chai, W. K., He, D., Psaras, I., and Pavlou, G. (2012). Cache “less for more” in information-centric networks. In *NETWORKING 2012*, pages 27–40. Springer.
- Chapela, V., Criado, R., Moral, S., and Romance, M. (2015). Mathematical foundations: Complex networks and graphs (a review). In *Intentional Risk Management through Complex Networks Analysis*, pages 9–36. Springer.
- Foster, K. C., Muth, S. Q., Potterat, J. J., and Rothenberg, R. B. (2001). A faster katz status score algorithm. *Computational & Mathematical Organization Theory*, 7(4):275–285.
- Fricke, C., Robert, P., Roberts, J., and Sbihi, N. (2012). Impact of traffic mix on caching performance in a content-centric network. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 310–315. IEEE.
- Grando, F., Noble, D., and Lamb, L. C. (2016). An analysis of centrality measures for complex and social networks. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, pages 1–6. IEEE.
- Guan, J., Quan, W., Xu, C., and Zhang, H. (2012). The location selection for ccn router based on the network centrality. In *Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on*, volume 2, pages 568–582. IEEE.
- Kutscher, D., Eum, S., Pentikousis, K., Psaras, I., Corujo, D., Saucez, D., Schmidt, T., and Waehlich, M. (2014). Icn research challenges. *Work in progress*.
- Laoutaris, N., Che, H., and Stavrakakis, I. (2006). The lcd interconnection of lru caches and its analysis. *Performance Evaluation*, 63(7):609–634.
- Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.
- Rossi, D. and Rossini, G. (2012). On sizing ccn content stores by exploiting topological information. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 280–285. IEEE.
- Saino, L., Psaras, I., and Pavlou, G. (2014). Icarus: a caching simulator for information centric networking (icn). In *Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques, SIMUTOOLS '14, ICST, Brussels, Belgium, Belgium*. ICST.
- Statstutor (2017). Pearson’s correlation.
- Sun, Y., Fayaz, S. K., Guo, Y., Sekar, V., Jin, Y., Kaafar, M. A., and Uhlig, S. (2014). Trace-driven analysis of icn caching algorithms on video-on-demand workloads. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pages 363–376. ACM.
- You, K., Tempo, R., and Qiu, L. (2017). Distributed algorithms for computation of centrality measures in complex networks. *IEEE Transactions on Automatic Control*, 62(5):2080–2094.
- Yufei, C., Min, Z., and Muqing, W. (2016). A centralized control caching strategy based on popularity and betweenness centrality in ccn. In *Wireless Communication Systems (ISWCS), 2016 International Symposium on*, pages 286–291. IEEE.