

# Estimativa contínua e tolerante a faltas de funções distribuição cumulativa em redes de larga escala<sup>\*</sup>

Miguel Borges, Carlos Baquero, Paulo Jesus, and Paulo Almeida

DI/CCTC, Universidade do Minho, Braga, Portugal

**Resumo** Em ambientes descentralizados de larga escala como é o caso das redes de sensores sem fios, P2P e outras, a recolha de dados é praticável apenas se houver agregação dos mesmos. No estado da arte actual, as estratégias de resolução desta questão não são satisfatórias dado exigirem que os protocolos sejam reiniciados sempre que os valores iniciais mudam ou quando o rácio de entrada/saída de nodos é não nulo. Acresce que nenhuma estratégia satisfaz a estimativa de CDFs (funções de distribuição cumulativa). A abordagem apresentada neste trabalho mostra como é possível o cálculo de funções de distribuição cumulativa em redes distribuídas, permitindo o seguimento dinâmico dos valores amostrados, sem que seja necessário reiniciar o protocolo em condições adversas de funcionamento da rede (perda de mensagens, entrada/saída de nodos). Esta abordagem é baseada num protocolo de *averaging* tolerante a faltas. Com este trabalho pretende-se também contribuir com uma estratégia que reduz o custo de comunicação entre nodos, com base em decisões locais e sensível à taxa de variação de estimativas. Os resultados de simulação mostram a resiliência deste protocolo, permitindo a estimativa contínua de CDFs em presença de dinamismo (perda de mensagens, alteração de valor amostrado, *churn*). Mostram também a convergência rápida na estimativa de CDFs para diferentes configurações de rede.

## 1 Introdução

A capacidade de agregar dados é uma característica fundamental na concepção de sistemas de informação escaláveis, que permite a determinação de propriedades globais importantes de forma descentralizada para a coordenação de aplicações distribuídas, ou para fins de monitorização.

Agregados simples como mínimos/máximos, contagens, somas e médias foram já extensivamente estudados no passado. No entanto este tipo de agregados pode não ser suficiente para caracterizar distribuições de dados enviesadas e na presença de valores atípicos (*outliers*), sendo relevante a determinação de uma estimativa dos valores na rede (e.g. histograma, CDF).

Métricas como médias ou desvio padrão escondem em muitos casos alterações na propriedade monitorizada que são relevantes para decisões de controlo.

---

<sup>\*</sup> Este trabalho é parcialmente financiado pelo projecto FCT Castor (PTDC/EIA-EIA/104022/2008)

Por exemplo, numa rede P2P é útil saber o espaço em disco de todos os nodos da rede. Uma métrica escalar como a média, diz pouco sobre a distribuição do espaço em disco. No entanto, a caracterização do espaço em disco como uma distribuição permite perceber quais os nodos com ou sem espaço disponível. Um outro exemplo genérico é o da monitorização de grandezas que sigam distribuições assimétricas (monitorização numa rede de detecção de fogo, carga numa rede de processadores): a distribuição estatística dos atributos envolvidos não é satisfatoriamente descrito com, por exemplo uma média, dado que facilmente valores anómalos podem enviesar a métrica. Nestes casos é fundamental caracterizar os atributos como uma distribuição estatística.

Existem diversas estratégias para a determinação de distribuições estatísticas de dados. Uma proposta recente [9] refere atingir uma precisão nas estimativas melhor que em abordagens anteriores. No entanto, mostra limitações na tolerância a faltas e no suporte à monitorização contínua de propriedades.

Neste trabalho apresentamos uma estratégia rápida e robusta para a determinação de distribuições estatísticas em redes de larga escala, em concreto a função distribuição cumulativa, baseada na técnica de Flow Update [4], tolerante a faltas e que permite a monitorização contínua de atributos em ambientes dinâmicos (*churn* e alteração dos valores amostrados).

Este artigo está organizado da seguinte forma: o trabalho relacionado é exposto na segunda secção, apresentando algoritmos distribuídos que efectuem a agregação de dados. Na terceira secção mostramos o modelo usado na determinação de CDFs em redes distribuídas bem como as características de rede e estratégia de divulgação de mensagens. Apresentamos também os detalhes mais relevantes do algoritmo. Na quarta secção é apresentado o mecanismo de quiescência. Na penúltima secção são apresentadas as métricas usadas para aferir a qualidade do algoritmo, resultados de simulação e a sua discussão. A última secção traça algumas conclusões gerais sobre o trabalho e apresenta futuras direcções de investigação.

## 2 Trabalho relacionado

O trabalho desenvolvido no domínio da agregação de dados em redes de larga escala compreende técnicas de agregação como o TAG [8], Push-pull *gossiping* [3], Flow Update [4], DRG [1]. Todos têm em comum a capacidade de determinar escalares tais como: médias, contagem de nodos, e somas. No entanto, a maioria das abordagens apresentadas acima não são imunes à perda de mensagens dado que se baseiam no princípio de distribuição de fracções de massa por todos os nodos da rede e em caso de perda de mensagem (i.e. massa), nunca poderão convergir para a estimativa correcta. No caso do Flow Update, há distribuição de fluxos pelos nodos da rede, com preservação dos seus valores amostrados, sendo a massa global do sistema sempre preservada.

No que diz respeito às técnicas de agregação de dados em redes de larga escala capazes de estimar distribuições probabilísticas, destacam-se na literatura actual

o Q-digest [10], H-GAP [6], classificação de dados baseada em modelos Gaussian Mix [2][5] e o Adam2 [9].

Em algumas das abordagens [10][6] encontram-se limitações na robustez pelo facto de se impor uma topologia em árvore na rede. Esta restrição limita o algoritmo no sentido em que uma falha num nodo poderá comprometer o resultado obtido (perdendo-se informação da sub-árvore). No caso dos modelos Gaussian Mix [2][5] não é admissível a duplicação nem a perda de mensagens.

Em [9] é usada uma técnica de determinação de funções distribuição cumulativa (CDF) criando o particionamento da propriedade a monitorizar em pontos de amostragem. Os nodos contribuem para cada um desse pontos caso o seu valor amostrado seja igual ou inferior. Esse algoritmo usa Push-Pull *gossiping* [3] para estimar a fracção dos nodos associado a cada ponto de amostragem. Neste algoritmo, a rede arranca com um conjunto de instâncias com pontos de amostragem equi-espaçados e com mínimo e máximo da propriedade pré-definido. Para acompanhar alterações dos valores amostrados, a estratégia usada exige que o protocolo seja re-iniciado periodicamente. Para além disso, os pressupostos desta abordagem não admitem perda de mensagens e é também susceptível à duplicação de mensagens.

### 3 Estimativa de Função de Distribuição Cumulativa

#### 3.1 Modelo

Uma Função de Distribuição Cumulativa (ou *CDF* - Cumulative Distribution Function) define-se como a probabilidade de uma variável aleatória  $X$  com uma dada função de distribuição ocorrer num valor igual ou inferior a  $x$ ,  $F(x) = P(X \leq x)$ . Dado que se pretende a estimativa de *CDF* discreta, será contada a proporção de valores da propriedade em  $k$  intervalos. Assim, a *CDF* é discretizada num conjunto de  $k$  intervalos com fronteiras, ou pontos de amostragem  $l$ , tais que:  $CDF = \{P(x \leq l_0), P(x \leq l_1), \dots, P(x \leq l_k)\}$ , com  $l_0$  o valor mínimo observado e  $l_k$  o valor máximo observado. Denominamos os  $l_k$  pontos de amostragem como *labels* e estes identificam o intervalo. Em cada intervalo irá ser estimada a proporção do atributo que verifica a condição de o seu valor amostrado  $V_i$  ser igual ou inferior aos *labels*. O tamanho de todos os intervalos da *CDF* é obtido por  $(max - min)/k$ . O número de intervalos  $k$  poderá ter um valor máximo  $k_{max}$ .

Cada nodo da rede pretende estimar uma *CDF* de um dado atributo cujos valores pertencem a  $\mathbb{R}$ , sem qualquer conhecimento *à priori* sobre a gama do atributo a ser medido.

Para uma dado intervalo  $l$  da *CDF*, a sua estimativa de acumulação  $e$  é resultado da média dos nodos  $N$  cujo valor amostrado  $V_i$  satisfaz a condição de ser igual ou inferior ao label  $l$ :

$$\forall n \in N, \forall l \in Labels : e = \frac{\sum_i \langle 1 | V_i \leq l \rangle}{|N|} \quad (1)$$

O cálculo da média para cada um dos intervalos é feito de forma dinâmica com *Flow Update* [4]. Trata-se de um protocolo distribuído de *averaging* baseado no princípio de distribuição de massa usado noutros algoritmos [1][3].

Com o *flow update*, a estimativa para um dado intervalo da *CDF* resulta da diferença entre o valor amostrado  $V_i$  no nodo  $i$  e o somatório de contributos  $f$  (*fluxos*) dos nodos vizinhos  $j$ .

O modelo de rede base é o de um grafo  $D$  não orientado com topologia *random*, com vértices fixos e com média de três ligações com nodos vizinhos.

Todos os nodos da rede enviam mensagens de modo síncrono por rondas, seguindo o modelo descrito em [7], mas o algoritmo é facilmente adaptável ao modelo assíncrono. O envio de mensagens é feito por *broadcast* para o conjunto de nodos vizinhos.

### 3.2 Algoritmo

O algoritmo distribuído para a determinação da *CDF* inicia-se com a medição, para cada um dos nodos, do seu valor amostrado  $V_i$ . De seguida cria-se uma distribuição de dados inicial com *label*  $l_0 = V_i$  e de frequência 1, como está descrito no Algoritmo 1. Este estado inicial verifica a condição  $P(V_i \leq l_0)$ .

---

**Algorithm 1** Cálculo da função de distribuição inicial, independente dos labels existentes

---

**Require:**  $initialDD = \emptyset \wedge V_i \neq \perp \wedge labels = \{V_i\}$   
**for all**  $label \in labels$  **do**  
    **if**  $V_i \leq label$  **then**  
         $initialDD \leftarrow initialDD \cup (label, 1)$   
    **else**  
         $initialDD \leftarrow initialDD \cup (label, 0)$   
    **end if**  
**end for**

---

Em seguida o nodo cria as mensagens para envio aos nodos vizinhos. Estas são constituídas por um conjunto de estimativas  $E$  na forma de tuplos (*label*, frequência),  $E = \bigcup_{\forall k} \{(l_k, f_k)\}$  que definem a *CDF* estimada no nodo. A mensagem é também composta por um conjunto de tuplos (*label*, fluxo) para cada vizinho  $j$ ,  $F_j = \bigcup_{\forall k} \{(l_k, f_{jk})\}$  e também pelo mínimo e máximo locais. Dado tratar-se da ronda inicial,  $E$  possui apenas o elemento  $(V_i, 1)$  e os fluxos  $F$  para os nodos vizinhos  $j$  são nulos:  $\forall j : F_j = \{(V_i, 0)\}$

Para todas as mensagens recebidas no nodo  $i$ , este procede primeiro ao recálculo das suas *labels* por forma a incorporar os valores mínimo e máximo enviados pelos vizinhos (descrito no Algoritmo 2). Depois, recalcula a sua distribuição local de acordo com esses novos valores de *labels*. O recálculo de posicionamento dos *labels* só acontece se o valor de mínimo ou máximo recebidos forem diferentes dos locais.

---

**Algorithm 2** Detecção de alteração de mínimo e máximo

---

**Require:**  $(receivedMin \wedge receivedMax \wedge min \wedge max) \neq \emptyset$   
**if**  $receivedMin \leq min$  **then**  
     $min \leftarrow receivedMin$   
     $rangeChanged \leftarrow \mathbf{true}$   
**end if**  
**if**  $receivedMax \geq max$  **then**  
     $max \leftarrow receivedMax$   
     $rangeChanged \leftarrow \mathbf{true}$   
**end if**  
**if**  $receivedMin > min \vee receivedMax < max$  **then**  
     $neighborDataNeedsAdjustment()$   
**end if**

---

No caso de não se verificar a alteração de mínimo ou máximo, o nodo procede apenas ao ajuste dos valores de estimativa de *CDF* e fluxos de nodos vizinhos para valores locais, tal como está descrito no Algoritmo 3.

---

**Algorithm 3** Criação de *labels* equi-espaçados

---

**Require:**  $newLabels = \emptyset \wedge labels \neq \emptyset$   
**Require:**  $(min \wedge max) \neq \emptyset \wedge min \leq max$   
**if**  $min = max$  **then**  
     $newLabels \leftarrow newLabels \cup \{max\}$   
**else**  
     $labelWidth \leftarrow (max - min)/k$   
     $label \leftarrow labelWidth + min$   
    **while**  $label < max$  **do**  
         $newLabels \leftarrow newLabels \cup \{label\}$   
         $label \leftarrow label + labelWidth$   
    **end while**  
     $newLabels \leftarrow newLabels \cup \{max\}$   
**end if**

---

Após estas operações de ajuste de referência, os valores de fluxo, estimativa e *labels* serão usados para o cálculo da nova estimativa de *CDF*.

De um modo intuitivo, para cada intervalo  $k$ , cada nodo da rede cujo valor amostrado seja igual ou inferior ao *label* correspondente, contribui com o valor 1. Caso contrário, contribui com o valor 0. Através do *averaging*, os nodos com um valor amostrado nessas condições irão estimar a frequência relativa dos restantes nodos de acordo com a equação 1.

De acordo com a técnica de *Flow Update*, e para cada um dos  $k$  intervalos, a nova estimativa  $e_{ik}$  é calculada pela diferença entre o valor amostrado no nodo  $V_i$  e a soma do valor simétrico dos fluxos recebidos  $f_{jik}$  pelos nodos vizinhos  $j \in D_i$  com a soma das estimativas recebidas dos nodos vizinhos  $e_{jik}$ . Este resultado é

dividido pelo número de nodos vizinhos  $|D_i|$ , mais um:

$$\forall k \in K : e_{ik} \leftarrow \frac{(V_i - \sum_{j \in D_i} f_{jik}) + \sum_{j \in D_i} e_{jik}}{|D_i| + 1} \quad (2)$$

Após aplicar o cálculo da nova estimativa a todos os valores de intervalo, o nodo calcula para cada  $k$  intervalo o valor dos fluxos para devolver aos seus vizinhos, que resulta da soma do simétrico dos fluxos  $f_{jik}$  recebidos com a diferença entre a estimativa calculada  $e_{ik}$  e a estimativa recebida  $e_{jik}$ :

$$\forall k \in K : f_{ijk} \leftarrow -f_{jik} + (e_{ik} - e_{jik}) \quad (3)$$

## 4 Quiescência

A motivação para a implementação de um mecanismo de quiescência, para que cada nodo pare de transmitir activamente, prende-se com a necessidade de reduzir o volume de informação passada entre nodos da rede, uma vez atingido um determinado grau de precisão na estimativa da *CDF*.

Em determinadas condições cada nodo pode decidir localmente não enviar informação sobre a sua estimativa actual e respectivos fluxos para os nodos vizinhos, permitindo assim baixar o custo das comunicações.

A decisão local é tomada aferindo a taxa de variação entre estimativas consecutivas. O algoritmo de quiescência tem como parâmetros dois valores que definem os níveis de histerese (o limite de variação mínimo  $\varepsilon_{low}$  a partir da qual o nodo entra no estado quiescente  $\chi = 1$  e o limite máximo  $\varepsilon_{high}$  a partir do qual o nodo abandona o estado quiescente,  $\chi = 0$ ) e um terceiro parâmetro  $r_{max}$  que determina o número de ciclos (rondas) durante os quais as condições de quiescência devem verificar-se para que o nodo entre ou saia do estado quiescente.

A criação de um mecanismo de histerese evita oscilações no estado de quiescência assim que  $\varepsilon_{low}$  é ultrapassado. Para que o nodo retorne ao estado de “acordado” é necessário que a variação local ultrapasse o limiar superior  $\varepsilon_{high}$ .

A taxa de variação entre estimativas é dada por:

$$\Delta_t(l) = \frac{P(X \leq l)_t - P(X \leq l)_{t-1}}{P(X \leq l)_t} \quad (4)$$

De uma *CDF* numa ronda  $r$  é considerado o intervalo cujo  $\Delta_t$  é máximo:

$$\Delta_{tmax} = \max_{l \in Labels} (\Delta_t(l)) \quad (5)$$

Caso  $\Delta_{tmax}$  seja inferior ao limiar de erro mínimo  $\varepsilon_{low}$ , o nodo entra em quiescência ( $\chi = 1$ ). Neste estado o nodo continua a monitorizar variações do seu valor amostrado e a computar novas estimativas de acordo com essa variação e de acordo com mensagens de nodos vizinhos que possa receber. Caso o resultado de um par de estimativas consecutivas calculada em quiescência seja superior ao limiar de erro máximo estipulado  $\varepsilon_{high}$ , o nodo passa então a enviar de novo

mensagens para a vizinhança. Um nodo arranca o protocolo com estado inicial acordado,  $\chi = 0$ .

$$\chi \leftarrow f(\Delta_t, l, \chi) = \begin{cases} 1 & \text{if } \max_l(\Delta_t(l)) \leq \varepsilon_{low} \wedge \chi = 0, \\ 0 & \text{if } \max_l(\Delta_t(l)) \geq \varepsilon_{high} \wedge \chi = 1, \\ \chi & \text{otherwise.} \end{cases} \quad (6)$$

A quiescência da rede é assim determinada por variações locais. O mecanismo não resolve a possibilidade de sucessivos incrementos na diferença com valor inferior a  $\varepsilon_{high}$  quando está quiescente. Uma variação lenta faria com que o nodo não acordasse. Existem soluções para esta questão que serão consideradas em trabalho futuro.

## 5 Simulação

### 5.1 Medição do erro

São usados como critério de avaliação qualitativa do protocolo dois erros de ajuste entre a estimativa dos nodos e a distribuição estatística dos valores iniciais atribuídos aos nodos. Os erros de ajuste são calculados em todas as  $r$  rondas.

Um dos erros de ajuste é o teste estatístico Kolmogorov-Smirnov que corresponde ao intervalo que possui maior diferença absoluta entre as frequências da função distribuição cumulativa estatística de referência, com base numa visão global sobre o valor dos nodos, e a função cumulativa determinada num nodo  $n$  na ronda  $r$  em que os intervalos sejam idênticos, de acordo com a expressão:

$$KS_r^n = \max_{l \in Labels} |P(X \leq l)_r - P(X \leq l)_r^n| \quad (7)$$

O segundo erro usado na análise é a diferença de áreas, que corresponde ao somatório da diferença entre as frequências da função cumulativa dos valores da distribuição estatística de referência nos nodos e a função cumulativa determinada por *averaging* num nodo na ronda  $r$  em que os intervalos sejam idênticos. A diferença é calculada para todos os  $l$  labels, com  $l_{max}$  o número máximo de labels e com  $max$  e  $min$  os valores máximo e mínimo do conjunto dos valores da distribuição estatística inicial, de acordo com a expressão:

$$\Sigma \Delta_r^n = \frac{max - min}{l_{max}} \sum_{l \in Labels} |P(X \leq k)_r - P(X \leq k)_r^n| \quad (8)$$

Com vista a medir o desempenho da rede como um todo, foram definidos erros máximos e médios para ambas as medidas  $KS$  e  $\Sigma \Delta$ . O erro máximo de uma medida reporta-se ao nodo com o maior erro no conjunto de todos os nodos  $N$ . O erro médio de uma medida refere-se à média dos erros em todos os nodos da rede. Assim temos que o erro KS máximo, i.e., nodo  $n$  com o maior desvio num dos seus intervalos à distribuição estatística inicial:

$$KS_{maxr} = \max_{n \in N} (KS_r^n) \quad (9)$$

O erro KS médio em todos os nodos  $n$  da rede é:

$$KS_{\mu r} = \frac{1}{|N|} \sum_{n \in N} KS_r^n \quad (10)$$

O erro de diferença de áreas máximo, i.e., nodo  $n$  cuja diferença de áreas entre a função cumulativa estimada e a função cumulativa efectiva é maior para todos os nodos da rede é dado pela expressão:

$$\Sigma \Delta_{maxr} = \max_{n \in N} (\Sigma \Delta_r^n) \quad (11)$$

E a medida de média de diferença de áreas para toda a rede de nodos  $n$  é dada pela expressão:

$$\Sigma \Delta_{\mu r} = \frac{1}{|N|} \sum_{n \in N} \Sigma \Delta_r^n \quad (12)$$

## 5.2 Setup experimental e resultados

Tanto quanto nos é dado conhecer, este trabalho é pioneiro na agregação de distribuições tolerante a faltas em sistemas distribuídos. O algoritmo directamente concorrente (Adam2), é baseado no algoritmo push-pull levando, em caso de perda de mensagens, à convergência para uma estimativa errada.

Para avaliar a solução proposta foram efectuadas uma série de simulações com diferentes configurações. Os resultados de simulação correspondem à média de 30 simulações efectuadas para cada uma das condições de teste.

São mostrados no Gráfico 1(d), para uma rede de 1000 nodos, os diferentes erros medidos na estimativa da *CDF*. Na apresentação dos restantes resultados de simulação é usada a métrica de erro médio  $KS_{\mu}$ , que permite ter uma noção da diferença média absoluta entre a distribuição real e a distribuição estimada.

**Variação na topologia** A simulação de variação da topologia da rede (apresentada em 1(a)) mostra que a estimativa da *CDF* converge mais rapidamente em redes *random*. O resultado foi obtido para uma conectividade média de 3.

**Tamanho da rede** O erro de ajuste médio KS é maior quanto maior for o tamanho da rede (apresentado em 1(b)), logo a convergência da estimativa é mais lenta para redes com um maior número de nodos.

**Perda de mensagens** Na simulação de estimativa de *CDF* com perda de mensagens (no Gráfico 1(c)) nota-se uma convergência mais rápida na ausência de perdas até cerca da ronda 50. Desse ponto em diante, o facto de se introduzir perda de mensagens faz com que a convergência acelere. Conclui-se então que o comportamento do algoritmo não só é resiliente à perda de mensagens como também a presença de uma taxa de 10% de perdas melhora a convergência da estimativa. Este resultado é coerente com o apresentado em [4].

**Perturbação e recuperação** Apresenta-se no Gráfico 2(a) a resiliência da rede quando introduzidas perturbações nos valores da distribuição inicial. No

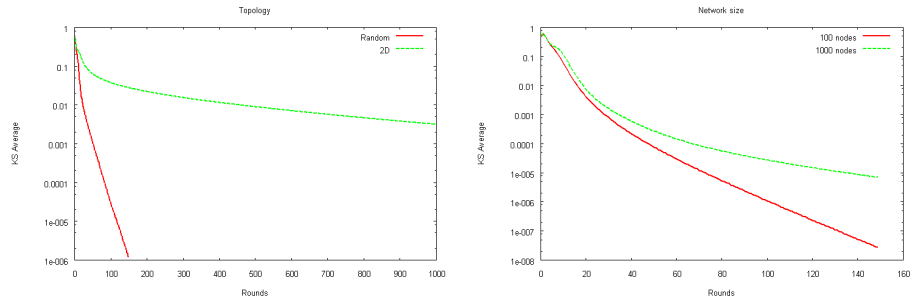


caso, um aumento no valor amostrado  $V_i$  de 10% em 20% dos nodos da rede na ronda 75. No momento da perturbação o erro médio total sobe para rapidamente convergir para um ajuste com um erro próximo de  $1e^{-5}$  na ronda 150. É mostrado em simultâneo o comportamento da rede sem perturbação.

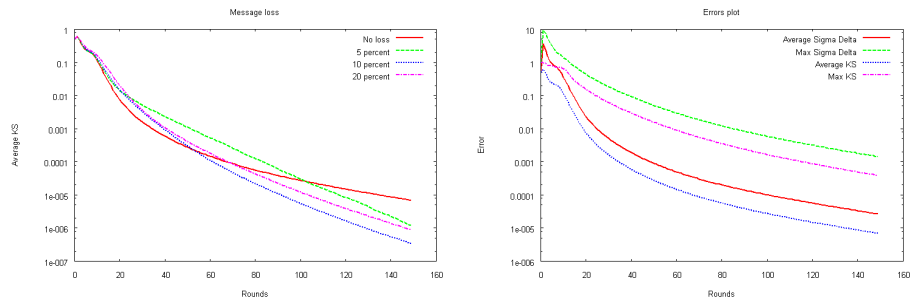
**Normal vs. exponencial** No que diz respeito ao comportamento da estimativa de  $CDF$  em presença de distribuições sintéticas diferentes (ver Gráfico 2(b)), nota-se uma convergência mais rápida na distribuição exponencial  $\lambda = 1.5$ . Este comportamento deve ser dependente do número de intervalos considerados (no caso  $k = 20$ ) pelo que um número de intervalos baixo e com uma distribuição exponencial com  $\lambda$  maior deverá apresentar uma convergência mais lenta. Esta possibilidade resulta do facto de as distribuições exponenciais “concentrarem” as frequências nos intervalos inferiores da propriedade.

**Entrada/saída de nodos** Um outro dado referente à resiliência do algoritmo é mostrado no Gráfico 2(c). Este apresenta a evolução do erro da estimativa para a introdução de 10% de nodos novos entre a roda 50 e a ronda 75. São depois removidos 10% de nodos na ronda 125. Estes resultados mostram a adaptabilidade do algoritmo a elevadas taxas de entrada/ saída de nodos e o quão rapidamente o algoritmo converge para valores de erro relativo médio baixos.

**Quiescência** No Gráfico 2(d) é apresentada uma simulação com o mecanismo de quiescência activado. Este foi parametrizado para um  $\varepsilon_{low} = \varepsilon_{high} = 0.01$ . No mesmo gráfico é apresentado o número cumulativo de mensagens trocadas entre os nodos. Pode observar-se que todos os nodos entram em estado quiescente cerca da ronda 45 e que até o número de mensagens trocadas vai diminuindo, acompanhando o adormecimento dos nodos, até cessar.



(a) Topologia 2D vs. random para uma rede de 1000 nodos (b) Tamanho da rede (100 e 1000 nodos)

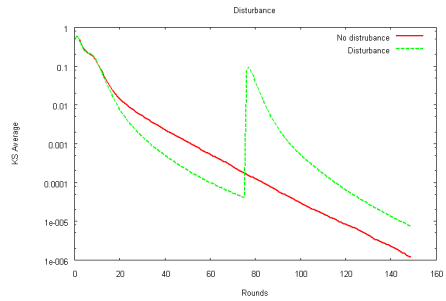


(c) Efeito de perda de mensagens na estimativa para uma rede random (d) Representação dos tipos de métrica de erro usados nas simulações

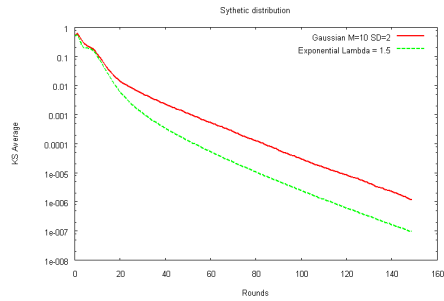
**Figura 1.** Resultados gráficos dos diferentes tipos de erro, topologias de rede, número de nodos e resiliência a perda de mensagens

## 6 Conclusões e trabalho futuro

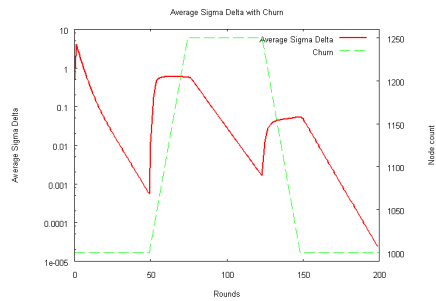
Apresentamos um algoritmo distribuído para a determinação de funções cumulativas de probabilidade em redes de larga escala. Este algoritmo tem como principais vantagens a imunidade à perda de mensagens e a relação logarítmica entre o tempo durante o qual o algoritmo corre e a precisão que se obtém na aproximação à distribuição original. É simultaneamente adaptável a alterações do valor amostrado e resiliente a dinamismo do número de nodos na rede. Permite também a utilização de um mecanismo de quiescência dos nodos assim que a variação local da estimativa é inferior a um determinado limiar, o que leva à diminuição do número de mensagens trocadas entre nodos. As distribuições determinadas por todos os nodos permitem a tomada de decisões que tirem partido do facto de se estar a agregar uma função probabilística: exclusão de *outliers* ou focada em quantis da propriedade observada. Por outro lado, cada nodo da rede possui uma estimativa global sobre o estado geral da propriedade distribuída, o que permite também a tomada de decisões com base em conhecimento local. Apresentamos também resultados de simulação que corroboram estas conclusões.



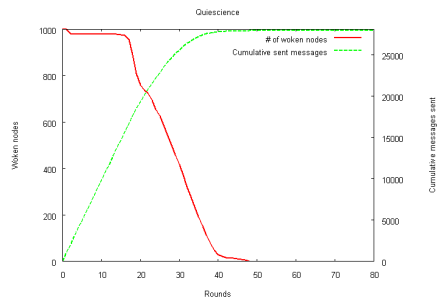
(a) Perturbação e recuperação numa rede random com 1000 nodos



(b) Normal vs. Exponencial



(c) Entrada/saída de nodos



(d) Quiescência vs. nro. de mensagens trocadas

**Figura 2.** Resultados gráficos que ilustram o comportamento do protocolo com dinamismo (diferentes distribuições estatísticas, churn, alterações dos valores amostrados) e com o mecanismo de quiescencia activado

Como trabalho futuro, pretende-se explorar a criação dinâmica de intervalos equi-frequentes por forma a aproximar os pontos de amostragem dos pontos onde a variação da função cumulativa é máxima. Irá ser também avaliado o custo de comunicação e processamento em comparação com sistemas semelhantes. Pretende-se ainda estudar o comportamento do algoritmo para determinar funções cumulativas de distribuições multivalor - em que cada nodo contribui com mais de um valor. Pretende-se ainda o tratamento das funções cumulativas com funções contínuas que permitirão um maior nível de compressão da CDF e testar um novo algoritmo baseado na manipulação de funções simbólicas. Pretende-se também analisar de que forma distribuições probabilísticas correlacionadas com a distribuição espacial dos nodos e a sua variação no tempo implicam na qualidade da determinação da função cumulativa. Poder-se-á ainda avaliar o modelo proposto num sistema real de monitorização.

## Referências

1. J.-Y. Chen, G. Pandurangan, and D. Xu. Robust computation of aggregates in wireless sensor networks: Distributed randomized algorithms and analysis. *Parallel and Distributed Systems, IEEE Transactions on*, 17(9):987–1000, sept. 2006.
2. Ittay Eyal, Idit Keidar, and Raphael Rom. Distributed data classification in sensor networks. In *PODC'10*, pages 151–160, 2010.
3. M. Jelasity and A. Montresor. Epidemic-style proactive aggregation in large overlay networks. In *Distributed Computing Systems, 2004. Proceedings. 24th International Conference on*, pages 102 – 109, 2004.
4. P. Jesus, C. Baquero, and P. Almeida. Fault-tolerant aggregation by flow updating. In Twittie Senivongse and Rui Oliveira, editors, *Distributed Applications and Interoperable Systems*, volume 5523 of *Lecture Notes in Computer Science*, pages 73–86. Springer Berlin / Heidelberg, 2009.
5. H. Jiang, S. Jin, and C. Wang. Parameter-based data aggregation for statistical information extraction in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 2010.
6. Dan Jurca and Rolf Stadler. H-gap: estimating histograms of local variables with accuracy objectives for distributed real-time monitoring. *IEEE Transactions on Network and Service Management*, 7:83–95, 2010.
7. Nancy A. Lynch. Distributed algorithms. pages 17–23. Morgan Kaufmann Publishers Inc., 1996.
8. Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.*, 36:131–146, December 2002.
9. Jan Sacha, Jeff Napper, Corina Stratan, and Guillaume Pierre. Adam2: Reliable distribution estimation in decentralised environments. *Distributed Computing Systems, International Conference on*, 0:697–707, 2010.
10. Nisheeth Shrivastava, Chiranjeev Buragohain, Divyakant Agrawal, and Subhash Suri. Medians and beyond: new aggregation techniques for sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 239–249, New York, NY, USA, 2004. ACM.