Protegendo BitTorrent: projeto e avaliação de contra-medidas eficazes para ataques DoS

Daniel Bauermann[1], Matheus Lehmann[1], Rodrigo Mansilha[1], Marinho P. Barcellos[2,3]

[1]Universidade do Vale do Rio dos Sinos (UNISINOS)
[2]Universidade Federal do Rio Grande do Sul (UFRGS)
[3]Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

Gramado, 04 de setembro de 2008

Roteiro

P2PSEC

Roteiro

P2PSEC

BitTorrent

- BitTorrent é um sistema de compartilhamento de arquivos P2P
- Tornou-se um padrão *de facto* para distribuição de arquivos
- Protocolo vulnerável a ataques de DoS

P2PSEC

Ataques ao BitTorrent

- Literatura apresenta trabalhos que exploram vulnerabilidades do BitTorrent
- Trabalhos anteriores visam maximizar recursos recebidos e minimizar recursos contribuídos
- Neste trabalho o objetivo do par malicioso é apenas prejudicar enxame

P2PSEC

Roteiro

P2PSEC

Visão geral

- Conteúdo: arquivos de dados que estão sendo compartilhados
- Conteúdo é divido em peças (e peças em blocos)
- Rastreador: elemento central que permite encontro de pares
- .torrent: arquivo com metadados sobre o contéudo e com informações do rastreador
- Semeadores: pares que possuem uma cópia completa dos dados
- Sugadores: pares que não são semeadores

P2PSEC

Visão orientada a conjuntos

BitTorrent

- b_x = bloco de uma peça (x)
- h_x = *hash* de uma peça (x)
- p_i = par local ($p_j, p_k, \ldots$ pares remotos)
- A_i = conjunto de pares ativos

P2PSEC

Ataques

Tipos de ataques:

- Eclipse
- Mentira de Peças
- Corrupção de Peças
- *Mentira em Massa*

Roteiro

Visão geral

- Um par p_i busca identificar os pares que constantemente permanecem “inativos”
- Tais pares são considerados “suspeitos” e temporariamente desconectados (colocados em quarentena Q)
- As conexões podem ser reutilizadas para conectar com outros pares
- r_j denota a taxa de troca de dados com par p_j
- Executa periodicamente

P2PSEC

Algoritmo Rotação de Pares

1: **for all** $p_j \in Q$ **do**
2: $q_j \leftarrow q_j - 1$
3: **if** $q_j = 0$ **then**
4: $Q \leftarrow Q \setminus \{p_j\}$
5: **end if**
6: **end for**
7: $a \leftarrow |P \setminus (A \cup Q)|$
8: **for all** $p_j \in A$, ordenado por r_j **do**
9: **if** $(t_j \geq t_{min} \wedge \frac{d_j}{t_j} < r_{min}) \wedge (|A| > A_{min} \vee (|A| \geq \lfloor \frac{3}{4} A_{min} \rfloor \wedge a > 0))$ **then**
10: $A \leftarrow A \setminus \{p_j\}$
11: $Q \leftarrow Q \cup \{p_j\}$
12: $q_j \leftarrow \lfloor cq_j \rfloor$
13: $cq_j \leftarrow cq_j \times f$
14: **if** $|A| < A_{min}$ **then**
15: $a \leftarrow a - 1$
16: **end if**
17: **end if**
18: **end for**
19: **while** $|A| < A_{min} \wedge P \setminus (A \cup Q) \neq \emptyset$ **do**
20: $p_k \leftarrow \forall p_j \in P \setminus (A \cup Q)$
21: $A \leftarrow A \cup \{p_k\}$
22: $t_k \leftarrow 0$
23: $d_k \leftarrow 0$
24: **end while**

P2PSEC

Algoritmo Rotação de Pares

1: **for all** $p_j \in Q$ **do**
2: $q_j \leftarrow q_j - 1$
3: **if** $q_j = 0$ **then**
4: $Q \leftarrow Q \setminus \{p_j\}$
5: **end if**
6: **end for**

- Verificação de pares que podem ser liberados da quarentena

P2PSEC

Algoritmo Rotação de Pares

7: $a \leftarrow |P \setminus (A \cup Q)|$
8: **for all** $p_j \in A$, ordenado por r_j **do**
9: **if** $(t_j \geq t_{min} \wedge \frac{d_j}{t_j} < r_{min}) \wedge (|A| > A_{min} \vee (|A| \geq \lfloor \frac{3}{4} A_{min} \rfloor \wedge a > 0))$ **then**
10: $A \leftarrow A \setminus \{p_j\}$
11: $Q \leftarrow Q \cup \{p_j\}$
12: $q_j \leftarrow \lfloor cq_j \rfloor$
13: $cq_j \leftarrow cq_j \times f$
14: **if** $|A| < A_{min}$ **then**
15: $a \leftarrow a - 1$
16: **end if**
17: **end if**
18: **end for**

- Análise de pares com menores contribuições, colocando-os em quarentena

P2PSEC

Algoritmo Rotação de Pares

19: **while** $|A| < A_{min} \wedge P \setminus (A \cup Q) \neq \emptyset$ **do**
20: $p_k \leftarrow \forall p_j \in P \setminus (A \cup Q)$
21: $A \leftarrow A \cup \{p_k\}$
22: $t_k \leftarrow 0$
23: $d_k \leftarrow 0$
24: **end while**

- Conecta-se com novos pares potencialmente contribuidores

P2PSEC

Roteiro

P2PSEC

Visão geral

- Tenta identificar pares que enviaram blocos corrompidos de uma peça
- Para tal emprega-se uma estratégia baseada em reputação
- Ao atingir um limiar mínimo p_j é desconectado de p_i e colocado em quarentena
- Algoritmo executado sempre que uma peça é completada

P2PSEC

Algoritmo Anti-corrupção

1: **if** $h_x = \text{hash}(b_x)$ **then**
2: **for all** $p_i \in R$ **do**
3: $o_i \leftarrow min(o_i + \delta_{inc}, 1)$
4: **end for**
5: **else**
6: $b'_x \leftarrow b_x$
7: $R \leftarrow U_x$
8: $B \leftarrow \{b_{x,y} | u_{x,y} \neq p_c\}$
9: **while** $\neg h'_x \wedge B \neq \emptyset \wedge Unchoked(p_c)$ **do**
10: $b'_{x,y} \leftarrow$ requests any $b'_{x,y} \in B$ from p_c
11: $B \leftarrow B \setminus \{b'_{x,y}\}$
12: **end while**
13: **if** $h'_x = \text{hash}(b'_x)$ **then**
14: $o_c \leftarrow min(o_c + \delta_{inc}, 1)$
15: $R' \leftarrow \{u_{x,y} | b_{x,y} \neq b'_{x,y}\}$
16: **for all** $p_i \in R'$ **do**
17: $o_i \leftarrow max(o_i - \delta_{dec}, 0)$
18: **end for**
19: $b_x \leftarrow b'_x$
20: **else**
21: **if** $p_c \notin A \vee \neg Unchoked(p_c)$ **then**
22: $o_c \leftarrow max(o_c - \frac{\delta_{dec}}{2}, 0)$
23: $b_x \leftarrow b'_x$
24: **else**
25: $o_c \leftarrow max(o_c - 2 \times \delta_{dec}, 0)$
26: **end if**
27: **end if**
28: **for all** $p_i \in R$ **do**
29: **if** $o_i = 0$ **then**
30: $A \leftarrow A \setminus \{p_c\}$
31: $Q \leftarrow Q \cup \{p_i\}$
32: **end if**
33: **end for**
34: **end if**

P2PSEC

Algoritmo Anti-corrupção

```
1: if h_x = hash(b_x) then
2:     for all p_i ∈ R do
3:         o_i ← min(o_i + δ_inc, 1)
4:     end for
5: else
```

- Se *hash* correto, incrementa reputação do(s) par(es) participante(s) da peça

P2PSEC

Algoritmo Anti-corrupção

```
5: else
6:     b'_x ← b_x
7:     R ← U_x
8:     B ← {b_{x,y} | u_{x,y} ≠ p_c}
9:     while ¬h'_x ∧ B ≠ ∅ ∧ Unchoked(p_c) do
10:        b'_{x,y} ← requests any b'_{x,y} ∈ B from p_c
11:        B ← B \ {b'_{x,y}}
12:    end while
```

- Se falha *hash*, copia peça e solicita blocos do par que contribuiu com último bloco

P2PSEC

Algoritmo Anti-corrupção

13: **if** $h_x^{'} = \text{hash}(b_x^{'})$ **then**
14: $o_c \leftarrow min(o_c + \delta_{inc}, 1)$
15: $R^{'} \leftarrow \{u_{x,y} | b_{x,y} \neq b_{x,y}^{'}\}$
16: **for all** $p_i \in R^{'}$ **do**
17: $o_i \leftarrow max(o_i - \delta_{dec}, 0)$
18: **end for**
19: $b_x \leftarrow b_x^{'}$
20: **else**

- Recuperando a peça com sucesso, incrementa reputação do par a contribuir por último e decrementa a reputação dos demais

P2PSEC

Algoritmo Anti-corrupção

20: **else**
21: **if** $p_c \notin A \vee \neg Unchoked(p_c)$ **then**
22: $o_c \leftarrow max(o_c - \frac{\delta_{dec}}{2}, 0)$
23: $b_x \leftarrow b_x^{'}$
24: **else**
25: $o_c \leftarrow max(o_c - 2 \times \delta_{dec}, 0)$
26: **end if**
27: **end if**

- Falhando na segunda tentativa, decrementa reputação do par que estava contribuindo por último

P2PSEC

Algoritmo Anti-corrupção

```
28:    for all p_i ∈ R do
29:      if o_i = 0 then
30:        A ← A \ {p_c}
31:        Q ← Q ∪ {p_i}
32:      end if
33:    end for
34: end if
```

- Pares com limite mínimo de reputação são desconectados e colocados em quarentena

Roteiro

Objetivos

Q1 Impacto dos ataques de mentira em massa e corrupção de peças

Q2 Eficácia das contra-medidas

Q3 Eficiência das contra-medidas

Q4 Quando e quão correto é a identificação dos maliciosos

P2PSEC

Modelo

Geral

- 1 semeador inicial
- 250 sugadores
- 64 peças
- peças 1 MB (64 blocos)

Mentira em massa

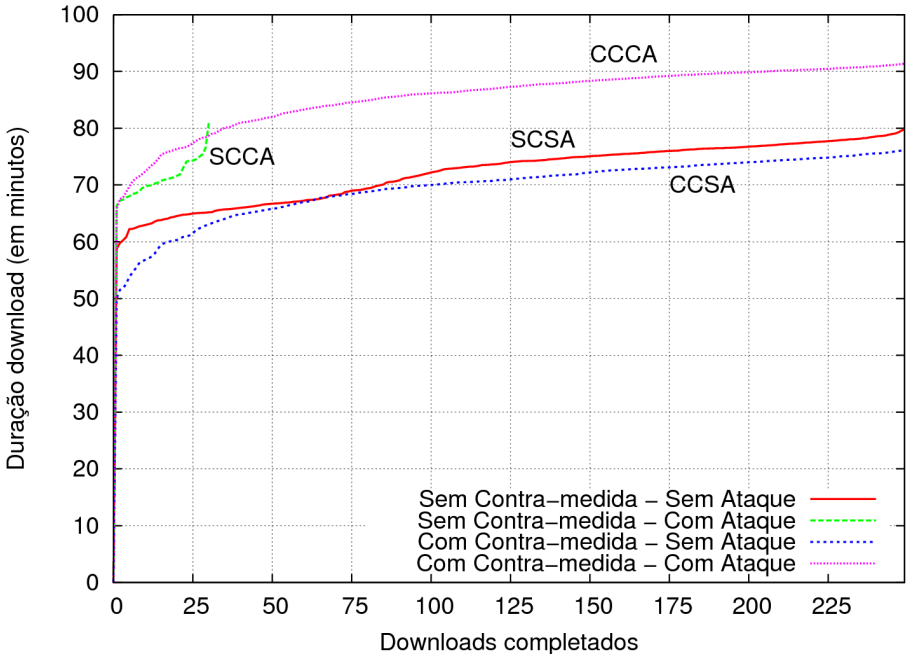
- 500 atacantes (*sybils*)
- 16 peças mentidas

Corrupção

- 15 atacantes
- 4s intervalos entre UNCHOKE

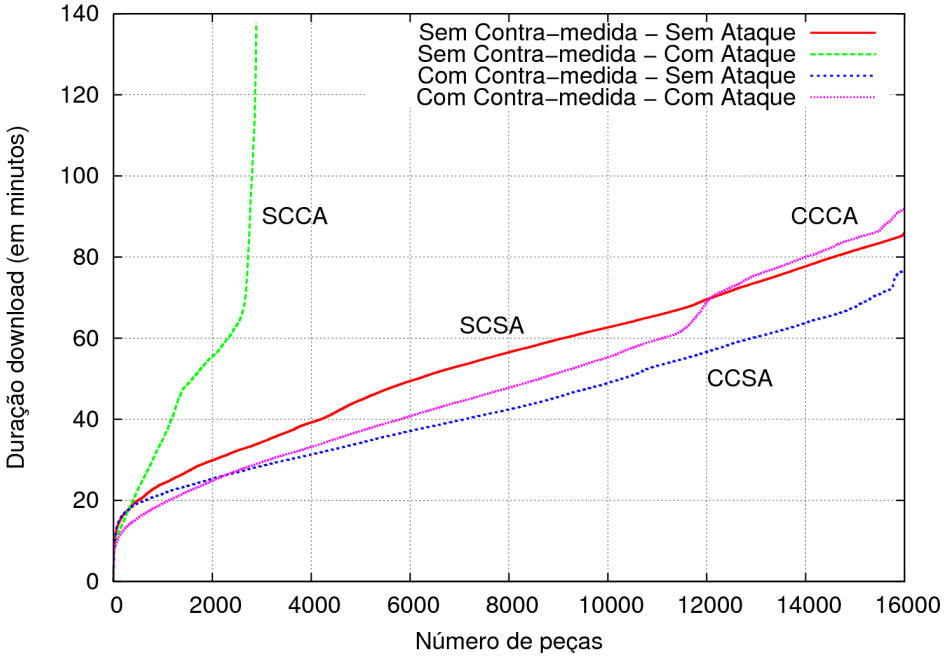
P2PSEC

Rotação de Pares - downloads



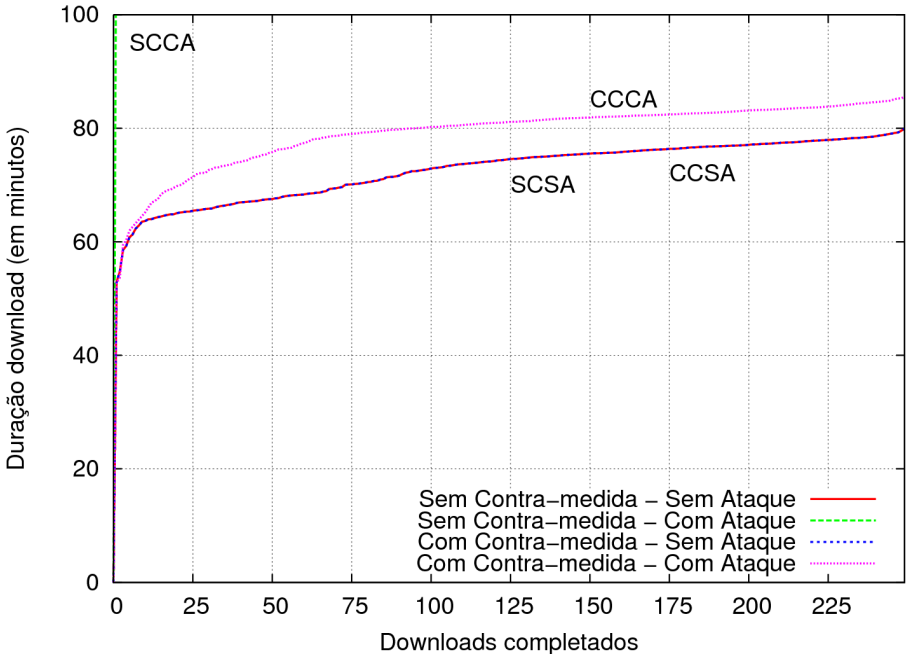
- Impacto ataque e eficácia contra-medida

Rotação de Pares - peças



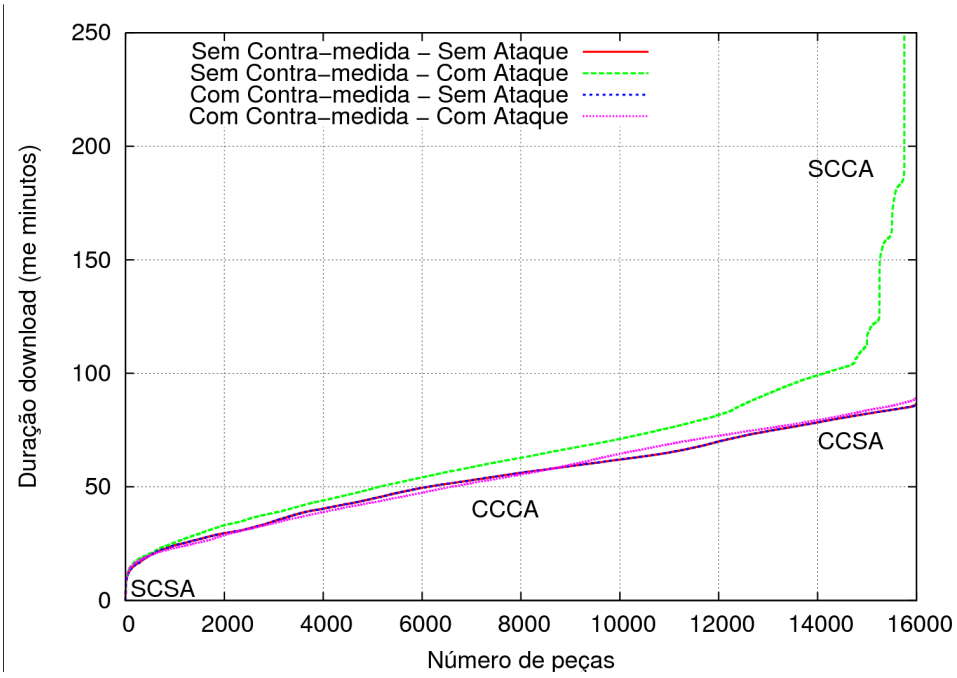
- Impacto ataque e eficácia contra-medida

Anti-corrupção - downloads



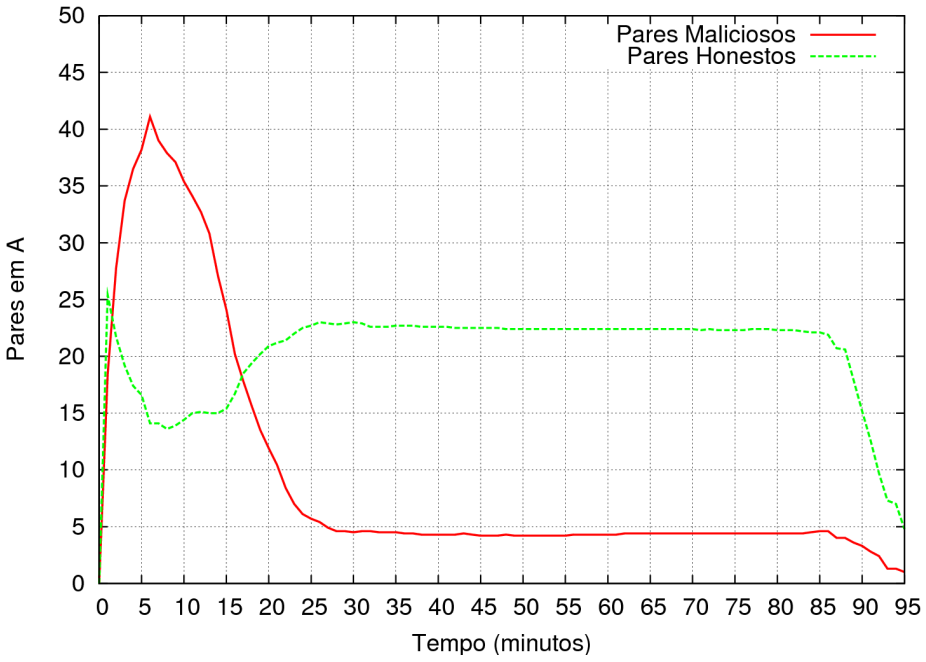
- Alto impacto ataque; sem ataque, sem e com contra-medida são idênticos (eficiência); e eficácia contra-medida

Anti-corrupção - peças



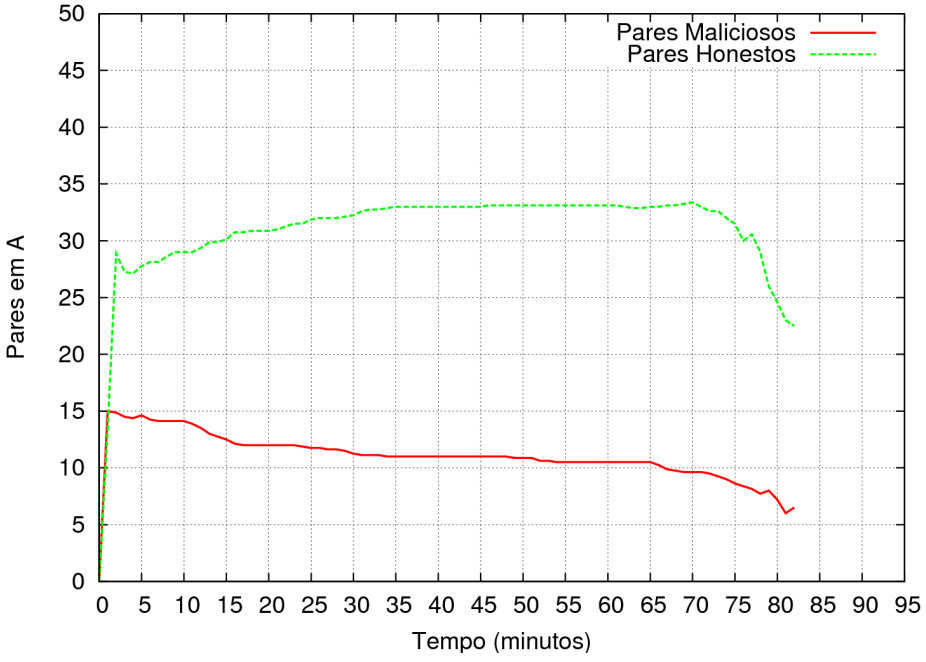
- Alto impacto ataque; sem ataque, sem e com contra-medida são idênticos (eficiência); e eficácia contra-medida

Detecção pares maliciosos - rotação



- Pares maliciosos crescem inicialmente; após 30 min média maliciosos é 5 maliciosos

Detecção pares maliciosos - corrupção



- Pares maliciosos começam a ser detectados após 3 min; nenhum par honesto é colocado em quarentena

Roteiro

Conclusões

- Algoritmos de contra-medidas mostraram-se eficazes e eficientes em combater ataques

Trabalhos Futuros

- Combinação de contra-medidas
- Avaliação experimental de algoritmos de contra-medidas

Protegendo BitTorrent: projeto e avaliação de contra-medidas eficazes para ataques DoS

Daniel Bauermann[1], Matheus Lehmann[1], Rodrigo Mansilha[1], Marinho P. Barcellos[2,3]

[1]Universidade do Vale do Rio dos Sinos (UNISINOS)
[2]Universidade Federal do Rio Grande do Sul (UFRGS)
[3]Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

Gramado, 04 de setembro de 2008

P2PSEC