Instituto Tecnológico de Aeronáutica

Divisão de Ciência da Computação
Departamento de Teoria da Computação
CTC-15: Inteligência Artificial

Trabalho de Exame

Planet Wars

Alunos
Cesar R. Kawakami
Guilherme R. N. Souza
Leonardo Ribeiro Carvalho
Rodrigo Simões Almeida

Professor
Carlos Henrique Ribeiro

02 de Dezembro de 2010

1 Objetivos

Fazer um jogador autônomo para Planet Wars, jogo usado na competição do Google Ai-Contest do ano de 2010. Esse jogador deve aplicar os conceitos e algoritmos vistos na matéria de Inteligência Artificial.

2 Introdução

Planet Wars é um jogo entre dois ou mais jogadores, cujo objetivo final é ter a maior frota de naves dentre todos os participantes. Apesar de uma partida poder ter vários jogadores, todas as partidas realizadas, tanto nos testes como na competição em si, foram realizadas com dois adversários.

O jogo é dividido em turnos no qual ambos jogadores tem todas informações relevantes disponíveis para tomar suas decisões e enviar frotas para combate. O combate entre frotas ocorre nos planetas, cada nave destrói exatamente uma nave inimiga e também é destruída. Ao final do combate quem tiver o maior número de naves sobreviventes no planeta o domina. Se restarem 0 naves o dono do planeta não muda. O jogador com planetas sob seu domínio tem sua frota local aumentada de acordo com a taxa de crescimento específica, a qual depende do tamanho do planeta dominado.

A abordagem para construção do jogador autônomo (a partir de agora denominado *bot*) dividiu-se em duas partes. A primeira foi o desenvolvimento de um *bot* que sabe se comunicar com o juíz/servidor e implementa heurísticas (estratégias básicas relacionados ao entendimento do funcionamento geral do jogo). Exemplos de fatores levados em consideração na heurística são: tentar defender suas naves, destruir naves inimigas, benefício trazido por conquistar planetas, entre outros. Mais informações na seção 3.

Essa versão inicial foi colocada no servidor do Google Ai-Contest e já apresentou bons resultados. A heurística, contudo, dependia de um conjunto de trinta e dois pesos associados a diferentes situações de jogo. Esses pesos haviam sido escolhidos sem muito critério, apenas seguiram um lógica geral relacionada principalmente ao sinal e módulo do peso associado. Para conseguir um melhor conjunto de pesos foi utilizado um algoritmo genético de evolução também detalhado na seção 3.

3 Implementação

3.1 *Bot* com estratégia básica

Inicialmete foi implementado um *bot* com estratégia básica. Toda a avaliação do estado do jogo considera um certo número de jogadas a frente, chamado horizonte. Esse horizonte funciona como uma poda de considerações muito distantes que poderiam atrasar o jogador e evita considerações que não fazem sentido, pois a previsão não considera que haverão jogadas, i.e., essa aboradagem avalia o jogo daqui a um número de jogadas se todos os jogadores tivessem como suas próximas ações não fazer nada. Esse horizonte é calculado levando em consideração

a distância média entre planetas a serem defendidos pelo *bot* e os planetas dominados pelo adversário.

A estratégia de jogo pode ser definida pelos passos:

- Para cada planeta acha-se quantas naves podem ser mandadas para o ataque sem que o planeta fique muito desprotegido e possa ser facilmente conquistado. Para isso, analisa-se as frotas que o estão atacando, achando quantas naves de defesa ele perderia a cada turno até o horizonte. O valor restante de naves ao atingir o horizonte é o valor que é considerado disponível para ataque e defesa, um valor negativo significa perda do domínio.
- Cada planeta com naves de crédito tenta defender outros planetas. Cada planeta pertencente ao jogador busca outros planetas que serão consquistados pelo inimigo (previsão negativa no item anterior) tentando fornecer naves suficientes para evitar essa perda. Essa ajuda a planetas vizinhos está condicionada à manutenção do domínio do planeta de origem das naves, caso a ajuda signifique perder o planeta que ajudou a ajuda não acontecerá. A ajuda também considera o tempo que a ajuda demorará para chegar ao planeta ajudado, se ele for conquistado antes de ser possível chegar a ajuda ela não será enviada.
- Com as naves restantes, após defender a si mesmo e fornecer ajuda a planetas aliados, cada planeta busca entre os planetas neutros e do inimigo aquele que apresenta o maior ganho esperado e realiza o ataque. Para cada planeta, é calculado um valor que representa o lucro esperado por atacá-lo. Dentro desse lucro é levado em consideração naves perdidas no ataque, naves perdidas pelo inimigo, quantas naves seriam ganhas e quantas naves o inimigo deixaria de ganhar devido à taxa de crescimento fornecida pelo planeta.

Essa estratégia considera os pontos tidos como essenciais para o bom desempenho do jogador e leva em consideração diversos jogos assistidos. A partir desta versão é possível fazer variantes jogarem entre si e selecionar a melhor dentre as opções geradas.

3.2 Algoritmo genético

4 Resultados e Análise

4.1 Definição de variáveis

A fim de facilitar a codificação das regras, algumas variáveis foram definidas.

```
export MY_IP=161.24.5.18
export LOOPBACK=127.0.0.0/8
export EXTERNAL_INT=eth0
export CLASS_A=10.0.0.0/8
export CLASS_B=172.16.0.0/12
export CLASS_C=192.168.0.0/16
export INTERNAL_NET=161.24.5.0/24
```

4.2 Iniciando os chains

Os seguintes comandos inicializam o `iptables`, com regras inicialmente limpas.

```
iptables -F   # limpa regras
iptables -X   # deleta cadeias
iptables -Z   # reseta contagens
```

Com regras inicialmente limpas, é natural que todos os tipos de conexões possam ocorrer normalmente. Os seguintes testes confirmam a situação.

Serviço	Origem	Destino	Status
SSH (TCP 22)	Este	Outro	OK
SSH (TCP 22)	Outro	Este	Erro[a]
FTP (TCP 21)	Este	Outro	OK
Ping (ICMP)	Este	Outro	OK
Ping (ICMP)	Outro	Este	OK

[a]Na máquina testada, o serviço sshd não estava disponível. Sendo assim, os testes de SSH de entrada não puderam ser executados com significância, e são omitidos pelo resto da experiência.

4.3 Definindo políticas padrão

Foram consideradas, neste passo, as seguintes regras.

- A política padrão seja rejeição de pacotes de entrada;
- A partir de sua máquina só sai ping e ftp para a servidora de ftp da sala (161.24.5.100).

As políticas padrão aplicadas, então, seguem.

```
iptables -P OUTPUT DROP
iptables -P INPUT DROP
iptables -P FORWARD DROP
```

Nesta situação, como não existe regra de aceitação de conexão, todos os testes falham.

Serviço	Origem	Destino	Status
SSH (TCP 22)	Este	Outro	Falha
FTP (TCP 21)	Este	Outro	Falha
Ping (ICMP)	Este	Outro	Falha
Ping (ICMP)	Outro	Este	Falha

4.4 Regras contra spoofings e flags

As seguintes regras bloqueiam alguns tipos de tentativas de spoofings e ataques de flags defeituosos.

```
# Recusar pacotes indicando virem da propria maquina
iptables -A INPUT -i $EXTERNAL_INT -s $MY_IP -j DROP

# Recusar pacotes indicando vir de redes privativas
iptables -A INPUT -i $EXTERNAL_INT -s $CLASS_A -j DROP
iptables -A INPUT -i $EXTERNAL_INT -s $CLASS_B -j DROP
iptables -A INPUT -i $EXTERNAL_INT -s $CLASS_C -j DROP

# Situacoes invalidas de flags TCP
iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
iptables -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
iptables -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
iptables -A INPUT -p tcp --tcp-flags FIN,RST FIN,RST -j DROP
```

Não foram testadas tentativas de ataque na rede. As regras acima, portanto, não foram testadas.

4.5 Regras de conexões estabelecidas

Utilizando-se do fato de que o `iptables` é um firewall *stateful*, isto é, que guarda o estado das conexões, pode-se declarar como válidos os pacotes pertencentes a conexões estabelecidas ou a novas conexões relacionadas a conexões já estabelecidas. A seguinte regra aplica o conceito.

```
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Como ainda não existem regras de aceitação de novas conexões, todos os testes ainda falham.

Serviço	Origem	Destino	Status
SSH (TCP 22)	Este	Outro	Falha
FTP (TCP 21)	Este	Outro	Falha
Ping (ICMP)	Este	Outro	Falha
Ping (ICMP)	Outro	Este	Falha

4.6 Regras desejadas

As seguintes regras foram consideradas.

- Sua máquina só aceite pedidos de conexão para o serviço de SSH;
- Pacotes UDP e ICMP também não sejam aceitos na entrada;

- Os pacotes rejeitados sejam descartados e logados;
- A partir de sua máquina só sai Ping e FTP para a servidora de FTP da sala (161.24.5.100).

Com base nas regras consideradas, o seguinte conjunto de regras foi elaborado.

```
# Liberar acesso ssh vindo da intranet
iptables -A INPUT -s $INTERNAL_NET -p tcp --dport ssh -j ACCEPT

# Liberar acesso PING e FTP para a servidora de FTP da sala
FTP_SERVER=161.24.5.100
iptables -A OUTPUT -d $FTP_SERVER -p tcp --dport ftp -j ACCEPT
iptables -A OUTPUT -d $FTP_SERVER -p icmp --icmp-type echo-request -j ACCEPT

# Logar demais pacotes (serao rejeitados)
iptables -A INPUT -j LOG
iptables -A OUTPUT -j LOG
iptables -A FORWARD -j LOG
```

O bloqueio de pacotes UDP e ICMP acontece como consequência direta da política de não-aceitação de pacotes.

Nesta situação, os testes apresentam os seguintes resultados.

Serviço	Origem	Destino	Status
SSH (TCP 22)	Este	Outro	Falha
FTP (TCP 21)	Este	161.24.5.100	Falha parcial
Ping (ICMP)	Este	Outro	Falha
Ping (ICMP)	Este	161.24.5.100	OK
Ping (ICMP)	Outro	Este	Falha

As mensagens logadas pelo `iptables` nas situações de falha são apresentadas a seguir.

```
# Tentando dar SSH em 161.24.5.19
[ 7124.302534] IN= OUT=eth0 SRC=161.24.5.18 DST=161.24.5.19 LEN=868 TOS=0x10\
 PREC=0x00 TTL=64 ID=63878 DF PROTO=TCP SPT=56986 DPT=22 WINDOW=2003 RES=0x00\
 ACK PSH URGP=0

# Tentando pingar 161.24.5.19
[ 7150.487894] IN= OUT=eth0 SRC=161.24.5.18 DST=161.24.5.19 LEN=84 TOS=0x00\
 PREC=0x00 TTL=64 ID=0 DF PROTO=ICMP TYPE=8 CODE=0 ID=62214 SEQ=2

# Tentando receber ping de 161.24.5.19
[ 7199.549871] IN=eth0 OUT= MAC=... SRC=161.24.5.19 DST=161.24.5.18 LEN=84\
 TOS=0x00 PREC=0x00 TTL=64 ID=7940 PROTO=ICMP TYPE=8 CODE=0 ID=42252 SEQ=0
# O MAC foi omitido por clareza e espaco.

# FTP em 161.24.5.100 ao dar um ls
```

```
[ 7348.783215] IN=eth0 OUT= MAC=... SRC=161.24.5.100 DST=161.24.5.18 LEN=60\
 TOS=0x00 PREC=0x00 TTL=64 ID=25515 DF PROTO=TCP SPT=20 DPT=45868 WINDOW=5840\
 RES=0x00 SYN URGP=0
```

Os resultados demonstram a manipulação de estado que o `iptables` fornece. Como o firewall detecta pacotes pertencentes a conexões estabelecidas, mesmo com o pacote `echo-reply` não estando explicitamente autorizado a entrar, o ping é bem sucedido pois a resposta é corretamente identificada como conexão já estabelecida. Outros pacotes de ping são corretamente rejeitados.

Um fenômeno curioso acontece com o serviço de FTP na configuração estabelecida. Enquanto a conexão inicial com o servidor e trocas de comandos são bem sucedidas, qualquer tentativa de transmissão de dados (mesmo um `ls`) falha, sendo bloqueada pelo firewall. Ocorre que o protocolo FTP, para correta manipulação no `iptables`, requer configuração mais sofisticada, uma vez que o protocolo abre uma conexão diferente exclusivamente para a transmissão de dados, conexão essa que o `iptables` não consegue, sozinho, identificar como parte do FTP[1].

5 Conclusão

A configuração de firewalls requer bom planejamento e atenção meticulosa. A definição de políticas de segurança depende de múltiplas variáveis e fatores como o equilíbrio entre usabilidade e segurança, sendo um trabalho complexo. Ademais, a atenção a detalhes é importante, uma vez que o firewall lida com os protocolos em baixo nível e não abstrai suas idiossincrasias.

[1] Para correta manipulação, o ideal seria carregar o módulo *ip_conntrack_ftp*, que permitiria que as conexões adicionais do protocolo FTP fossem acompanhadas pelo firewall e autorizadas por meio do filtro de estado RELATED.