

ACH2147 - Desenvolvimento de Sistemas de Informação Distribuídos

Aula 14 – Abordagens de Sincronização

Norton Trevisan Roman

23 de junho de 2022

Abordagens de Sincronização

- Algoritmos de Eleição
- Sistemas de Localização

Abordagens de Sincronização

- **Algoritmos de Eleição**
- Sistemas de Localização

Algoritmos de Eleição

- Muitos algoritmos distribuídos precisam que um processo assuma o papel de coordenador
 - A questão é como selecionar esse processo especial **dinamicamente**?

Algoritmos de Eleição

- Muitos algoritmos distribuídos precisam que um processo assuma o papel de coordenador
 - A questão é como selecionar esse processo especial **dinamicamente**?
- Em muitos sistemas o coordenador é escolhido manualmente (ex: servidores de arquivos)
 - Isso leva a soluções centralizadas com um ponto único de falha

Algumas questões...

- Se um coordenador é escolhido dinamicamente, até que ponto podemos dizer que o sistema será centralizado e não distribuído?

Algumas questões...

- Se um coordenador é escolhido dinamicamente, até que ponto podemos dizer que o sistema será centralizado e não distribuído?
- Um sistema inteiramente distribuído (sem um coordenador) é sempre mais robusto que uma solução centralizada/coordenada?

Pressupostos básicos

- Todo processo P possui um id único $id(P)$

Pressupostos básicos

- Todo processo P possui um id único $id(P)$
- Todos os processos conhecem os ids de todos os outros processos no sistema
 - Contudo, não têm como saber se eles estão rodando ou não

Algoritmos de Eleição

Pressupostos básicos

- Todo processo P possui um id único $id(P)$
- Todos os processos conhecem os ids de todos os outros processos no sistema
 - Contudo, não têm como saber se eles estão rodando ou não
- Em geral, os algoritmos de eleição tentam localizar o processo de maior id rodando em um dado momento
 - Designando-o como coordenador

O algoritmo do *bully*

- Considere N processos
 - $\{P_0, \dots, P_{N-1}\}$, com $id(P_k) = k$

Algoritmos de Eleição

O algoritmo do *bully*

- Considere N processos
 - $\{P_0, \dots, P_{N-1}\}$, com $id(P_k) = k$
- Quando algum processo percebe que o coordenador não está mais respondendo a requisições, ele começa uma eleição

Algoritmos de Eleição

O algoritmo do *bully*

- P_k inicia uma eleição da seguinte maneira:
 1. P_k envia uma mensagem ELECTION para todos os processos com identificadores maiores que o seu:
 - $P_{k+1}, P_{k+2}, \dots, P_{N-1}$
 2. Se ninguém responder, P_k ganha a eleição e se torna o coordenador
 3. Se um dos nós com maior id responder, esse assume
 - A eleição e o trabalho de P_k termina – ele desiste
 - O maior sempre ganha, por isso o nome de “algoritmo do valentão”

Algoritmos de Eleição

O algoritmo do *bully*

- A qualquer momento um processo P_k pode receber uma mensagem de eleição
- Este deve então responder com um OK, indicando que está vivo e irá assumir

Algoritmos de Eleição

O algoritmo do *bully*

- A qualquer momento um processo P_k pode receber uma mensagem de eleição
- Este deve então responder com um OK, indicando que está vivo e irá assumir
- P_k então faz uma nova eleição, a menos que já esteja fazendo alguma

Algoritmos de Eleição

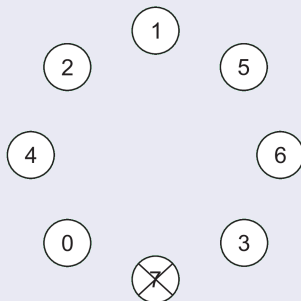
O algoritmo do *bully*

- A qualquer momento um processo P_k pode receber uma mensagem de eleição
 - Este deve então responder com um OK, indicando que está vivo e irá assumir
 - P_k então faz uma nova eleição, a menos que já esteja fazendo alguma
- No final, todos os processos irão desistir menos um
 - E este será o novo coordenador
 - Ele então anuncia sua vitória enviando uma mensagem aos demais

Algoritmos de Eleição

O algoritmo do *bully*: exemplo

Considere a rede de overlay ao lado, em que P_7 , o coordenador, não está mais respondendo

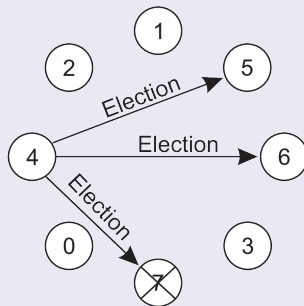


Algoritmos de Eleição

O algoritmo do *bully*: exemplo

Considere a rede de overlay ao lado, em que P_7 , o coordenador, não está mais respondendo

P_4 é o primeiro a perceber isso. Ele inicia então uma eleição, enviando ELECTION aos processos com maior id que o seu



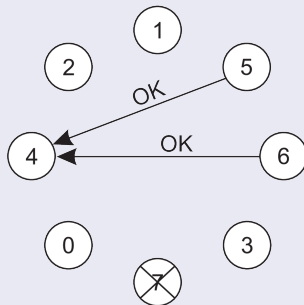
Algoritmos de Eleição

O algoritmo do *bully*: exemplo

Considere a rede de overlay ao lado, em que P_7 , o coordenador, não está mais respondendo

P_4 é o primeiro a perceber isso. Ele inicia então uma eleição, enviando ELECTION aos processos com maior id que o seu

P_5 e P_6 respondem com OK



Algoritmos de Eleição

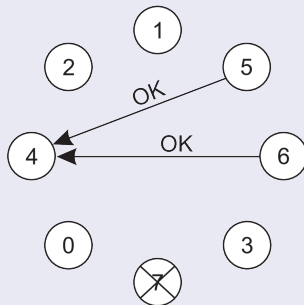
O algoritmo do *bully*: exemplo

Considere a rede de overlay ao lado, em que P_7 , o coordenador, não está mais respondendo

P_4 é o primeiro a perceber isso. Ele inicia então uma eleição, enviando ELECTION aos processos com maior id que o seu

P_5 e P_6 respondem com OK

Ao receber o primeiro OK, P_4 desiste – Um dos demais irá assumir



Algoritmos de Eleição

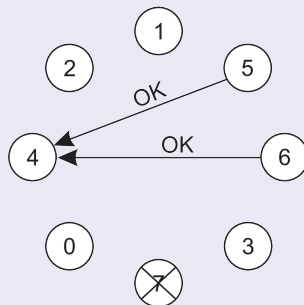
O algoritmo do *bully*: exemplo

Considere a rede de overlay ao lado, em que P_7 , o coordenador, não está mais respondendo

P_4 é o primeiro a perceber isso. Ele inicia então uma eleição, enviando ELECTION aos processos com maior id que o seu

P_5 e P_6 respondem com OK

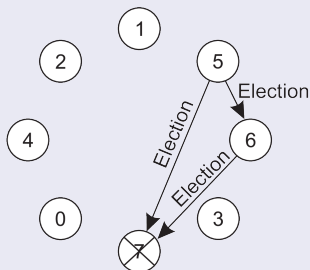
Ao receber o primeiro OK, P_4 desiste – Um dos demais irá assumir
Ele então aguarda a mensagem final que definirá o coordenador



Algoritmos de Eleição

O algoritmo do *bully*: exemplo

P_5 e P_6 fazem então uma eleição
cada um, enviando mensagem aos
processos com ids maiores que os seus

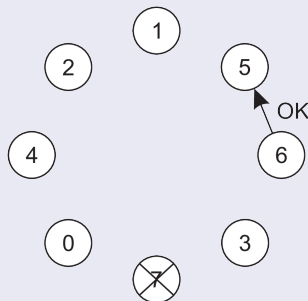


Algoritmos de Eleição

O algoritmo do *bully*: exemplo

P_5 e P_6 fazem então uma eleição
cada um, enviando mensagem aos
processos com ids maiores que os seus

P_6 diz a P_5 que irá assumir,
fazendo com que ele desista



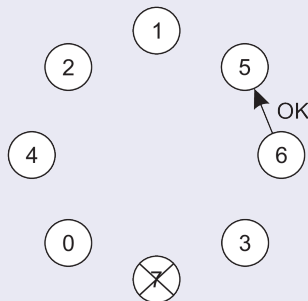
Algoritmos de Eleição

O algoritmo do *bully*: exemplo

P_5 e P_6 fazem então uma eleição
cada um, enviando mensagem aos
processos com ids maiores que os seus

P_6 diz a P_5 que irá assumir,
fazendo com que ele desista

Como não recebeu OK de ninguém,
 P_6 sabe que P_7 caiu e que ele venceu



Algoritmos de Eleição

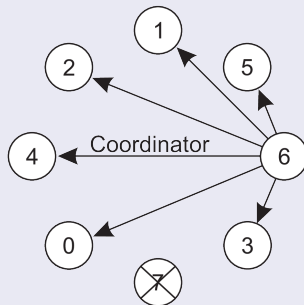
O algoritmo do *bully*: exemplo

P_5 e P_6 fazem então uma eleição cada um, enviando mensagem aos processos com ids maiores que os seus

P_6 diz a P_5 que irá assumir, fazendo com que ele desista

Como não recebeu OK de ninguém, P_6 sabe que P_7 caiu e que ele venceu

Quando estiver pronto para assumir, P_6 anuncia sua liderança a todos



Algoritmos de Eleição

O algoritmo do *bully*: exemplo

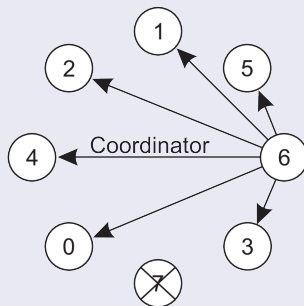
P_5 e P_6 fazem então uma eleição cada um, enviando mensagem aos processos com ids maiores que os seus

P_6 diz a P_5 que irá assumir, fazendo com que ele desista

Como não recebeu OK de ninguém, P_6 sabe que P_7 caiu e que ele venceu

Quando estiver pronto para assumir, P_6 anuncia sua liderança a todos

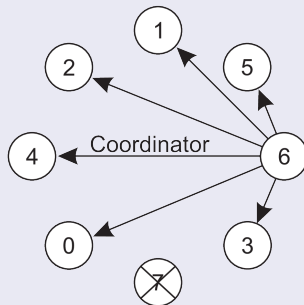
Estes podem então continuar seu trabalho, com P_6 como coordenador



Algoritmos de Eleição

O algoritmo do *bully*: exemplo

Se P_7 voltar à vida, ele envia a todos os demais uma mensagem COORDINATOR, forçando-os à submissão

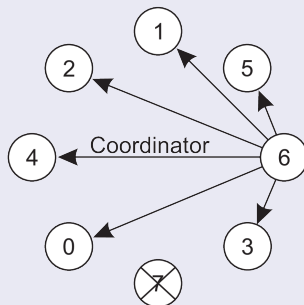


Algoritmos de Eleição

O algoritmo do *bully*: exemplo

Se P_7 voltar à vida, ele envia a todos os demais uma mensagem COORDINATOR, forçando-os à submissão

Cuidado! Assumimos aqui que a comunicação é confiável (não há perda de mensagem)



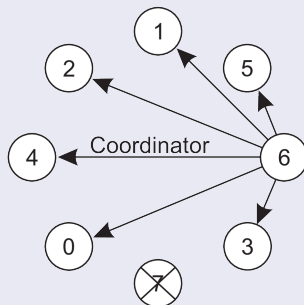
Algoritmos de Eleição

O algoritmo do *bully*: exemplo

Se P_7 voltar à vida, ele envia a todos os demais uma mensagem COORDINATOR, forçando-os à submissão

Cuidado! Assumimos aqui que a comunicação é confiável (não há perda de mensagem)

Esse algoritmo pode ser adaptado para, em vez do id, o vencedor ser o algoritmo com menor carga de trabalho (que deve ser comunicada junto à mensagem de eleição)



Algoritmos de Eleição

Algoritmo do anel

- A prioridade dos processos é obtida organizando-os em um anel lógico
 - O processo com maior prioridade deve ser eleito coordenador

Algoritmos de Eleição

Algoritmo do anel

- A prioridade dos processos é obtida organizando-os em um anel lógico
 - O processo com maior prioridade deve ser eleito coordenador
- Quando um processo nota que o coordenador caiu, constrói uma mensagem de eleição, contendo seu id
 - Ele envia então a mensagem ao seu sucessor
 - Se um sucessor estiver indisponível, a mensagem é enviada ao próximo sucessor, e assim por diante

Algoritmos de Eleição

Algoritmo do anel

- Se uma mensagem for repassada, o novo remetente adiciona seu id a ela, enviando-a ao seu sucessor
- Os ids são adicionados a uma lista, registrando por onde a mensagem passou
- Quando a mensagem volta ao nó que iniciou, todos tiveram a chance de anunciar a sua presença

Algoritmos de Eleição

Algoritmo do anel

- Se uma mensagem for repassada, o novo remetente adiciona seu id a ela, enviando-a ao seu sucessor
 - Os ids são adicionados a uma lista, registrando por onde a mensagem passou
 - Quando a mensagem volta ao nó que iniciou, todos tiveram a chance de anunciar a sua presença
- O nó inicial reconhece que a mensagem voltou quando vê seu id na lista de identificadores

Algoritmo do anel

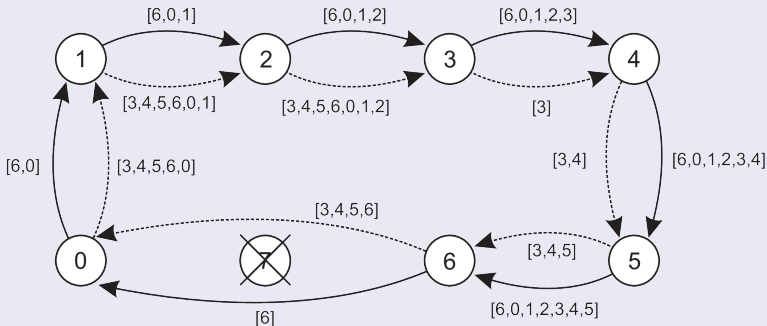
- O tipo da mensagem é então mudado para COORDINATOR e circulada novamente
 - Conterá assim uma lista dos processos vivos
 - Serve para informar a todos quem é o coordenador (o membro da lista com maior id) e quais são os membros do novo anel

Algoritmo do anel

- O tipo da mensagem é então mudado para COORDINATOR e circulada novamente
 - Conterá assim uma lista dos processos vivos
 - Serve para informar a todos quem é o coordenador (o membro da lista com maior id) e quais são os membros do novo anel
- Uma vez que essa mensagem tenha circulado, ela é removida, e todos voltam ao trabalho

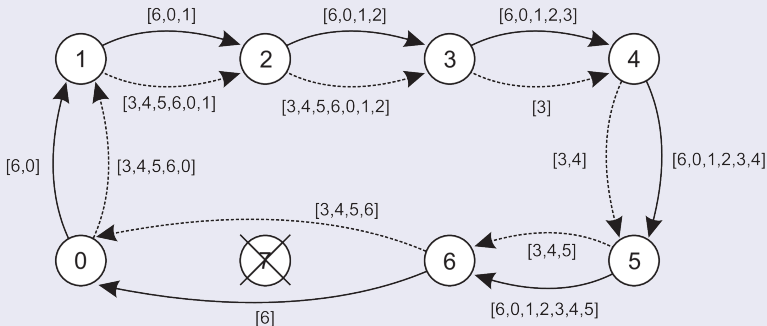
Algoritmos de Eleição

Algoritmo do anel: Exemplo



Algoritmos de Eleição

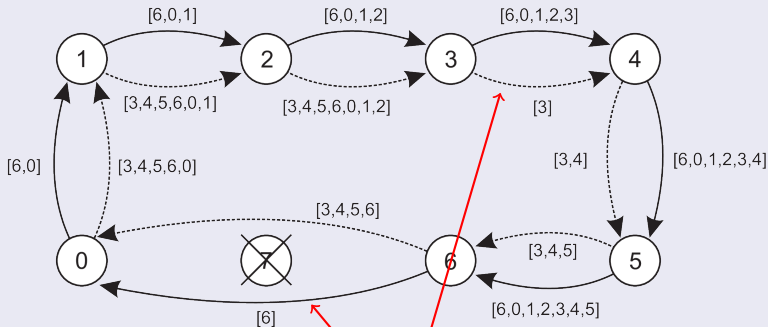
Algoritmo do anel: Exemplo



Suponha que P_3 e P_6 descubrem, ao mesmo tempo, que o coordenador P_7 caiu

Algoritmos de Eleição

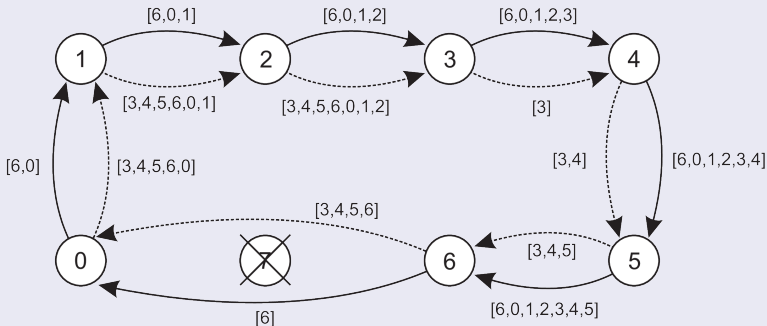
Algoritmo do anel: Exemplo



Cada um constrói então uma mensagem de eleição, começando sua circulação

Algoritmos de Eleição

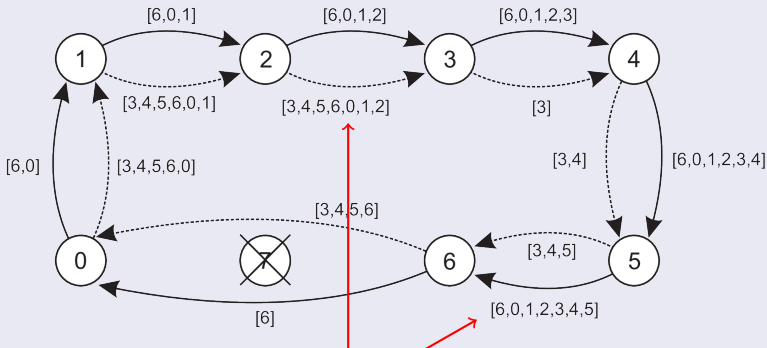
Algoritmo do anel: Exemplo



A cada nó que passa, o id desse nó é adicionado à mensagem

Algoritmos de Eleição

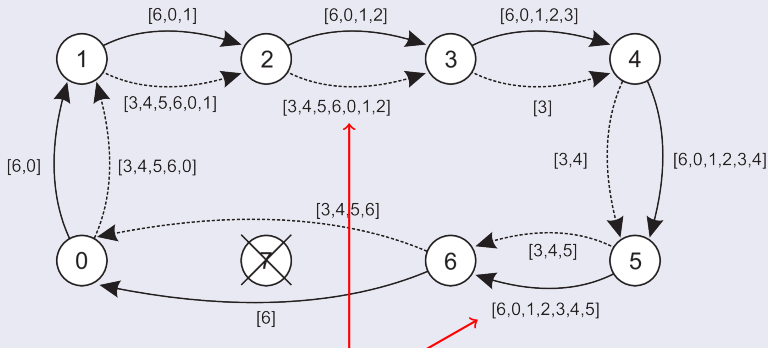
Algoritmo do anel: Exemplo



Ambas as mensagens acabarão por circular completamente. P_3 e P_6 as convertem então em COORDINATOR

Algoritmos de Eleição

Algoritmo do anel: Exemplo



Ambas conterão exatamente os mesmos membros em suas listas, na mesma ordem, avisando a todos sobre o vencedor

Eleição em redes sem fio

- Algoritmos de eleição tradicionais assumem que
 - A transmissão da mensagem é confiável
 - A topologia da rede não muda

Eleição em redes sem fio

- Algoritmos de eleição tradicionais assumem que
 - A transmissão da mensagem é confiável
 - A topologia da rede não muda
- Esse raramente é o caso com redes sem fio

Eleição em redes sem fio

- Algoritmos de eleição tradicionais assumem que
 - A transmissão da mensagem é confiável
 - A topologia da rede não muda
- Esse raramente é o caso com redes sem fio
- Considere uma rede sem fio em que os nós não se movem
 - Ainda assim, não necessariamente estão ao alcance de todos

Algoritmos de Eleição

Eleição em redes sem fio

- Para eleger um líder, cada nó pode iniciar uma eleição (este será o nó-fonte)
- Enviando uma mensagem ELECTION a seus vizinhos imediatos (no caso, os nós ao seu alcance)

Algoritmos de Eleição

Eleição em redes sem fio

- Para eleger um líder, cada nó pode iniciar uma eleição (este será o nó-fonte)
 - Enviando uma mensagem ELECTION a seus vizinhos imediatos (no caso, os nós ao seu alcance)
- Quando um nó R recebe uma mensagem ELECTION pela primeira vez, faz do remetente Q seu pai
 - Repassando então a mensagem a todos seus vizinhos imediatos, exceto seu pai
 - E esperando confirmações da mensagem chegarem, antes de confirmá-la a Q

Eleição em redes sem fio

- Quando um nó recebe uma ELECTION de alguém que não é seu pai, simplesmente confirma o recebimento
- Ele já repassou a mensagem aos vizinhos

Algoritmos de Eleição

Eleição em redes sem fio

- Quando um nó recebe uma ELECTION de alguém que não é seu pai, simplesmente confirma o recebimento
 - Ele já repassou a mensagem aos vizinhos
- Note que vizinhos de R que já selecionaram um pai enviarão rapidamente essa confirmação a ele
 - E se todos os vizinhos tiverem um pai, R é uma folha, e confirmará rapidamente a Q também
 - Ao fazer isso, também reportará informação, como tempo de bateria e outros recursos que tenha (sua **capacidade**)

Eleição em redes sem fio

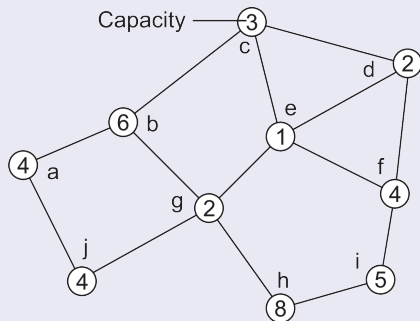
- Essa informação, por sua vez, permite a Q comparar a capacidade de R às dos outros nós
- Selecionando assim o melhor candidato para a liderança

Eleição em redes sem fio

- Essa informação, por sua vez, permite a Q comparar a capacidade de R às dos outros nós
- Selecionando assim o melhor candidato para a liderança
- Quando Q finalmente confirmar a seu pai P , enviará esse melhor nó também
- O nó-fonte acabará por saber qual nó é o melhor candidato para liderança, podendo então fazer o *broadcasting* dessa informação

Algoritmos de Eleição

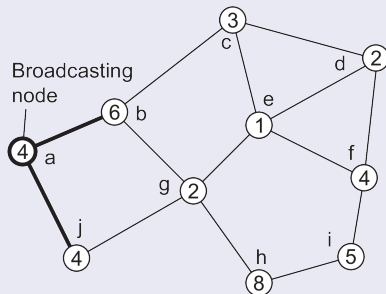
Eleição em redes sem fio: Exemplo



Algoritmos de Eleição

Eleição em redes sem fio: Exemplo

O nó *a* inicia a eleição,
fazendo o broadcast de
ELECTION a seus vizinhos

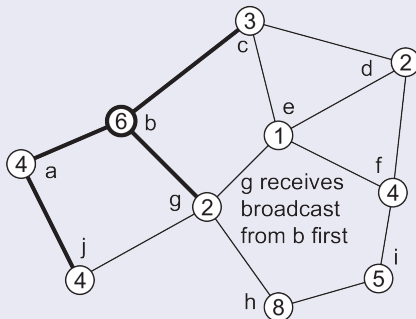


Algoritmos de Eleição

Eleição em redes sem fio: Exemplo

O nó *a* inicia a eleição,
fazendo o broadcast de
ELECTION a seus vizinhos

Mensagens ELECTION são
propagadas por toda a rede,
com cada nó marcando seu pai

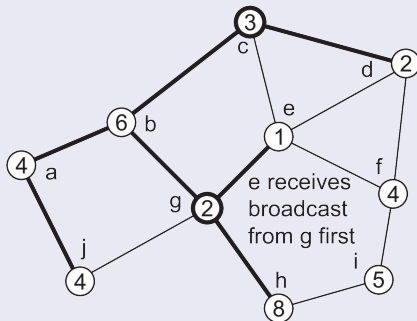


Algoritmos de Eleição

Eleição em redes sem fio: Exemplo

O nó *a* inicia a eleição,
fazendo o broadcast de
ELECTION a seus vizinhos

Mensagens ELECTION são
propagadas por toda a rede,
com cada nó marcando seu pai

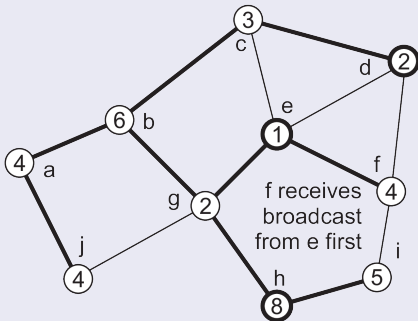


Algoritmos de Eleição

Eleição em redes sem fio: Exemplo

O nó *a* inicia a eleição,
fazendo o broadcast de
ELECTION a seus vizinhos

Mensagens ELECTION são
propagadas por toda a rede,
com cada nó marcando seu pai



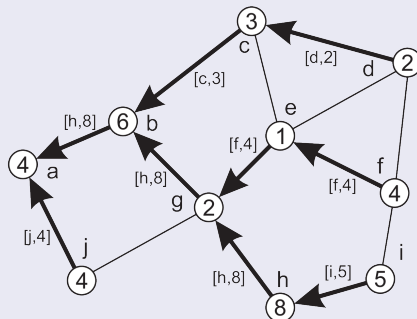
Algoritmos de Eleição

Eleição em redes sem fio: Exemplo

O nó *a* inicia a eleição,
fazendo o broadcast de
ELECTION a seus vizinhos

Mensagens ELECTION são
propagadas por toda a rede,
com cada nó marcando seu pai

Cada nó responde então a
seu pai, reportando o nó com
melhor capacidade que encontrou



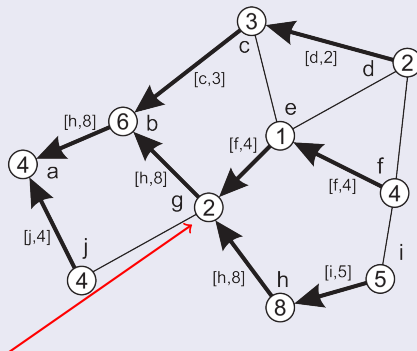
Algoritmos de Eleição

Eleição em redes sem fio: Exemplo

O nó *a* inicia a eleição,
fazendo o broadcast de
ELECTION a seus vizinhos

Mensagens ELECTION são
propagadas por toda a rede,
com cada nó marcando seu pai

Cada nó responde então a
seu pai, reportando o nó com
melhor capacidade que encontrou



Assim, quando *g* recebe as confirmações
de *e* e *h*, irá propagar *h* ao seu pai *b*

Algoritmos de Eleição

Eleição em redes sem fio

- Quando múltiplas eleições são iniciadas, cada nó deve escolher de qual participar

Eleição em redes sem fio

- Quando múltiplas eleições são iniciadas, cada nó deve escolher de qual participar
- Para isso, cada nó-fonte marca sua mensagem de eleição com um identificador

Eleição em redes sem fio

- Quando múltiplas eleições são iniciadas, cada nó deve escolher de qual participar
- Para isso, cada nó-fonte marca sua mensagem de eleição com um identificador
- Os nós participarão então da eleição com o maior identificador

Eleição em redes sem fio

- Quando múltiplas eleições são iniciadas, cada nó deve escolher de qual participar
 - Para isso, cada nó-fonte marca sua mensagem de eleição com um identificador
 - Os nós participarão então da eleição com o maior identificador
 - Interrompendo qualquer participação que tenham em outra eleição

Abordagens de Sincronização

- Algoritmos de Eleição
- **Sistemas de Localização**

Sistemas de Localização

Posicionando nós

Em SDs de grande escala, onde os nós estão espalhados ao longo de uma rede de área ampla (*wide-area network*), frequentemente precisamos levar em consideração a **proximidade** dos nós

Sistemas de Localização

Posicionando nós

Em SDs de grande escala, onde os nós estão espalhados ao longo de uma rede de área ampla (*wide-area network*), frequentemente precisamos levar em consideração a **proximidade** dos nós

Para isso, precisamos calcular a **localização** (relativa) de um nó

Sistemas de Localização

Calculando a posição

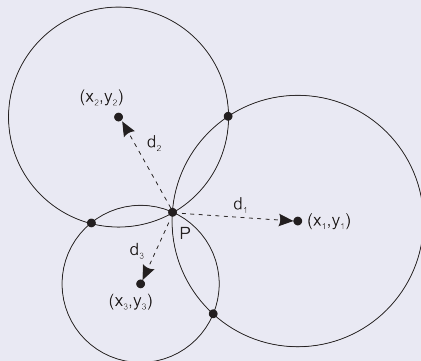
Em um espaço d -dimensional, um nó P precisa de $d + 1$ pontos de referência para calcular sua posição

Sistemas de Localização

Calculando a posição

Em um espaço d -dimensional, um nó P precisa de $d + 1$ pontos de referência para calcular sua posição

Considere o caso bidimensional



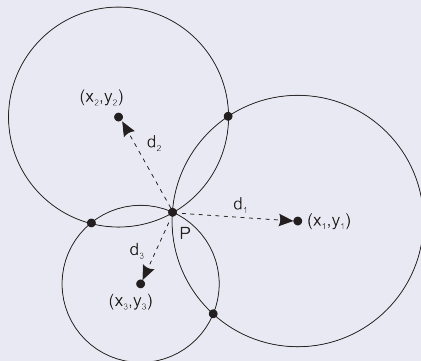
Sistemas de Localização

Calculando a posição

Em um espaço d -dimensional, um nó P precisa de $d + 1$ pontos de referência para calcular sua posição

Considere o caso bidimensional

Contatar um único nó diz a P
em qual círculo está localizado



Sistemas de Localização

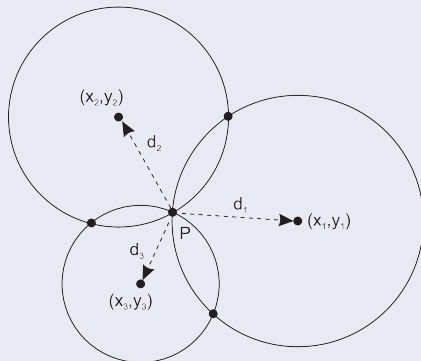
Calculando a posição

Em um espaço d -dimensional, um nó P precisa de $d + 1$ pontos de referência para calcular sua posição

Considere o caso bidimensional

Contatar um único nó diz a P
em qual círculo está localizado

Contatar dois nós diz a P
a interseção dos 2 círculos
(geralmente 2 pontos)



Sistemas de Localização

Calculando a posição

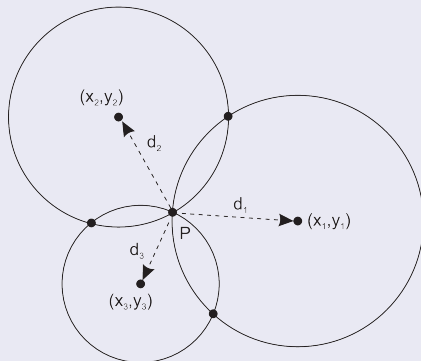
Em um espaço d -dimensional, um nó P precisa de $d + 1$ pontos de referência para calcular sua posição

Considere o caso bidimensional

Contatar um único nó diz a P
em qual círculo está localizado

Contatar dois nós diz a P
a interseção dos 2 círculos
(geralmente 2 pontos)

Um terceiro nó permitiria a P
calcular sua localização real

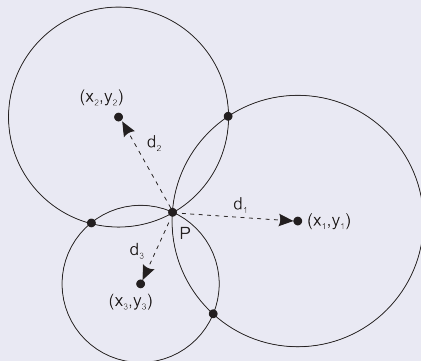


Sistemas de Localização

Calculando a posição

Em um espaço d -dimensional, um nó P precisa de $d + 1$ pontos de referência para calcular sua posição

A interseção dos 3 círculos é um ponto único



Sistemas de Localização

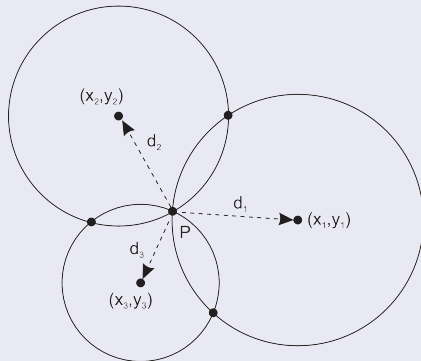
Calculando a posição

Em um espaço d -dimensional, um nó P precisa de $d + 1$ pontos de referência para calcular sua posição

A interseção dos 3 círculos é um ponto único

E, para conhecê-lo, precisamos resolver um sistema de 3 equações com 2 incógnitas (x_P, y_P) :

$$d_i = \sqrt{(x_i - x_P)^2 + (y_i - y_P)^2}$$



Global Positioning System (GPS)

- GPS é um SD baseado em satélites
 - Com coisa de 72 satélites, orbitando a aproximadamente 20.000 Km de altitude

Global Positioning System (GPS)

- GPS é um SD baseado em satélites
 - Com coisa de 72 satélites, orbitando a aproximadamente 20.000 Km de altitude
- Cada satélite tem até 4 relógios atômicos
 - Calibrados regularmente a partir de estações na Terra

Global Positioning System (GPS)

- GPS é um SD baseado em satélites
 - Com coisa de 72 satélites, orbitando a aproximadamente 20.000 Km de altitude
- Cada satélite tem até 4 relógios atômicos
 - Calibrados regularmente a partir de estações na Terra
- Os satélites continuamente fazem o broadcast de sua posição
 - Adicionando a cada mensagem seu horário local

Global Positioning System (GPS)

- Sabendo nossa distância a 4 satélites podemos então calcular nossa posição
 - Longitude, latitude e altitude

Global Positioning System (GPS)

- Sabendo nossa distância a 4 satélites podemos então calcular nossa posição
 - Longitude, latitude e altitude
- Contudo...
 - Leva algum tempo até que o sinal do satélite chegue ao receptor

Global Positioning System (GPS)

- Sabendo nossa distância a 4 satélites podemos então calcular nossa posição
 - Longitude, latitude e altitude
- Contudo...
 - Leva algum tempo até que o sinal do satélite chegue ao receptor
 - O relógio do receptor geralmente não está sincronizado com o do satélite

Global Positioning System (GPS)

- Sejam então...
 - Δ_r : **defasagem** (desconhecida) do relógio do receptor
 - x_r, y_r, z_r : **coordenadas** (desconhecidas) do receptor
 - T_i : timestamp da mensagem do satélite S_i
(supomos os T_i s precisos e sincronizados)
 - T_{agora} : tempo atual real
 - $\Delta_i = (T_{agora} - T_i) + \Delta_r$: atraso **medido pelo receptor**, entre envio e recebimento da mensagem
(O atraso real + a defasagem do relógio do receptor)

Global Positioning System (GPS)

- A distância real entre o receptor e S_i será
$$d_i = c \cdot (T_{agora} - T_i)$$
(uma vez que o sinal viaja à velocidade da luz (c))

Global Positioning System (GPS)

- A distância real entre o receptor e S_i será

$$d_i = c \cdot (T_{agora} - T_i)$$

(uma vez que o sinal viaja à velocidade da luz (c))

- Contudo, a distância **medida pelo receptor** a S_i será

$$\begin{aligned}\tilde{d}_i &= c \cdot \Delta_i \\ &= c \cdot [(T_{agora} - T_1) + \Delta_r] \\ &= c \cdot (T_{agora} - T_1) + c \cdot \Delta_r \\ &= d_i + c \cdot \Delta_r\end{aligned}$$

Global Positioning System (GPS)

- A distância real entre o receptor e S_i será

$$d_i = c \cdot (T_{agora} - T_i)$$

(uma vez que o sinal viaja à velocidade da luz (c))

- Contudo, a distância **medida pelo receptor** a S_i será

$$\begin{aligned}\tilde{d}_i &= c \cdot \Delta_i \\ &= c \cdot [(T_{agora} - T_1) + \Delta_r] \\ &= c \cdot (T_{agora} - T_1) + c \cdot \Delta_r \\ &= d_i + c \cdot \Delta_r\end{aligned}$$

E a distância real será $d_i = \tilde{d}_i - c \cdot \Delta_r$

Global Positioning System (GPS)

- Tomando
 - (x_i, y_i, z_i) como as coordenadas do satélite S_i , e
 - $d_i = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2}$ a distância real entre o receptor e o satélite

Global Positioning System (GPS)

- Tomando
 - (x_i, y_i, z_i) como as coordenadas do satélite S_i , e
 - $d_i = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2}$ a distância real entre o receptor e o satélite
- Temos que

$$\begin{aligned}\tilde{d}_i - c \cdot \Delta_r &= d_i \\ \tilde{d}_i - c \cdot \Delta_r &= \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2} \\ c \cdot \Delta_i - c \cdot \Delta_r &= \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2}\end{aligned}$$

Global Positioning System (GPS)

- $c \cdot \Delta_i - c \cdot \Delta_r = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2}$
- Temos um sistema de equações quadráticas com 4 incógnitas (x_r , y_r , z_r e Δ_r)

Global Positioning System (GPS)

- $c \cdot \Delta_i - c \cdot \Delta_r = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2}$
- Temos um sistema de equações quadráticas com 4 incógnitas (x_r , y_r , z_r e Δ_r)
- Precisamos então de 4 pontos de referência (satélites) para encontrar uma solução única

Global Positioning System (GPS)

- $c \cdot \Delta_i - c \cdot \Delta_r = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2}$
- Temos um sistema de equações quadráticas com 4 incógnitas (x_r , y_r , z_r e Δ_r)
- Precisamos então de 4 pontos de referência (satélites) para encontrar uma solução única
- Note que a medida do GPS nos dá também o valor do tempo real, ao nos dar Δ_r

Sistemas de Localização

Serviços de posicionamento via WiFi

- O maior problema com GPS é o fato de não funcionar bem em ambientes internos

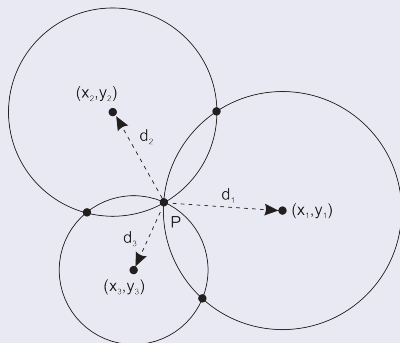
Serviços de posicionamento via WiFi

- O maior problema com GPS é o fato de não funcionar bem em ambientes internos
- Podemos, nesses casos, usar os pontos de WiFi disponíveis:
 - Assuma a existência de um banco de dados com as coordenadas de *access points* (APs) conhecidos
 - Assuma que podemos estimar a distância até um AP

Sistemas de Localização

Serviços de posicionamento via WiFi

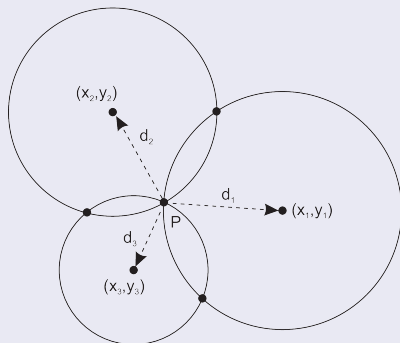
- Nessas condições, com três APs detectados, podemos calcular uma posição
- Lembre que estaremos em 2d



Sistemas de Localização

Serviços de posicionamento via WiFi

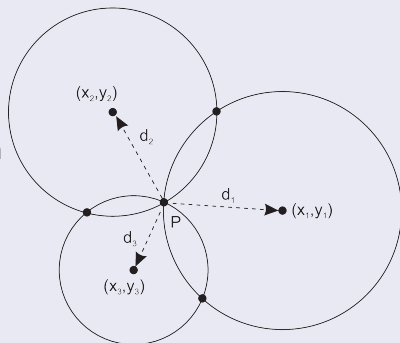
- Nessas condições, com três APs detectados, podemos calcular uma posição
- Lembre que estaremos em 2d
- A questão é: e se não tivermos tal base com as coordenadas dos APs?
- Como podemos localizá-los?



Serviços de posicionamento via WiFi

- **War driving**

- Use um dispositivo WiFi com um receptor GPS e se mova por uma área enquanto grava os pontos de acesso detectados

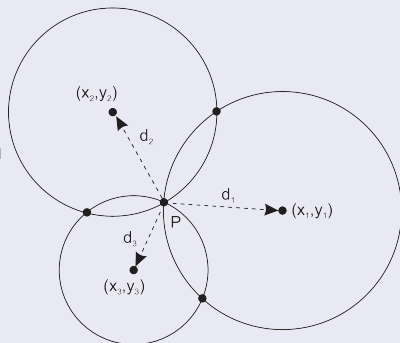


Sistemas de Localização

Serviços de posicionamento via WiFi

- **War driving**

- Use um dispositivo WiFi com um receptor GPS e se mova por uma área enquanto grava os pontos de acesso detectados
- Um AP pode ser identificado por seu SSID (*service set identifier*) ou *MAC address* (*media access control address*)

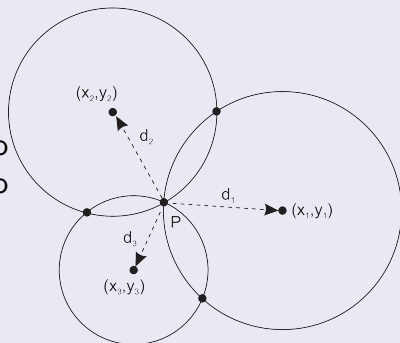


Sistemas de Localização

Serviços de posicionamento via WiFi

- Após detectar o AP em N locais diferentes $\{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$ (cuas coordenadas foram dadas pelo GPS), estime a localização do AP como sendo o centróide desse conjunto

$$\vec{x}_{AP} = \frac{\sum_{i=1}^N \vec{x}_i}{N}$$



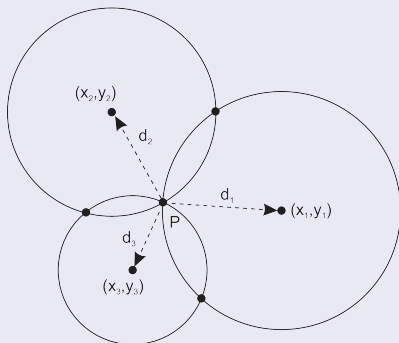
Sistemas de Localização

Serviços de posicionamento via WiFi

- Podemos melhorar a acurácia se levarmos em conta a força do sinal:

$$\vec{x}_{AP} = \frac{\sum_{i=1}^N f(AP) \cdot \vec{x}_i}{N}$$

- Damos assim mais peso a locais com sinal mais forte



Sistemas de Localização

Serviços de posicionamento via WiFi

- E qual o problema?
 - Acurácia limitada de cada ponto \vec{x}_i detectado pelo GPS
 - APs têm um alcance de transmissão não uniforme
 - O número de pontos da amostra (N) pode ser muito pequeno
 - Estimativas assim podem ficar dezenas de metros fora da localização real

