GBC074 – Sistemas Distribuídos

Comunicação – MOM e Protocolos Epidêmicos

MOM

- ***Message Oriented Middleware*** (MOM),
  - *Middlewares* focados nas mensagens trocadas entre processos em um nível mais alto do que *sockets*
- Diversas formas:
  - ***Message Passing Interface*** (MPI): usada em aplicações HPC (*high performance computing*)
  - ***Message Queues*** (MQ)
  - ***Publisher/Subscriber*** (PubSub): usados em sistemas de informação, ***Internet of Things*** (IOT) e ***Big Data***.

MPI

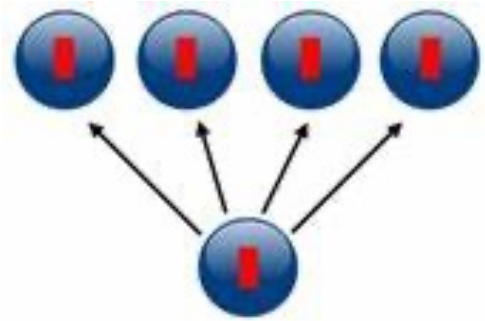
- ***Message Passing Interface:***
  - Usada para coordenar a distribuição e agregação de dados em aplicações em HPC (*high performance computing*)
  - Implementações se concentram em torno das linguagens percebidas como de melhor desempenho e mais usadas pelas comunidades que fazer uso de HPC, como C, C++ e Fortran
  - Exemplo: ***OpenMPI***
    - Código livre e bem mantida pela sua comunidade, é focada nestas três linguagens
    - Há também uma versão para Java

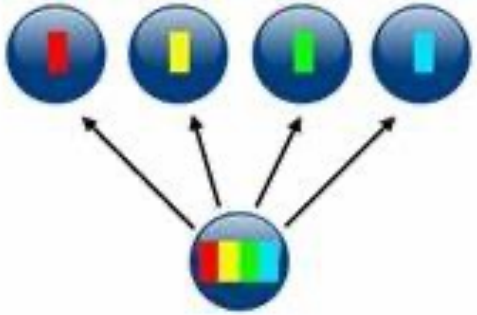
MPI

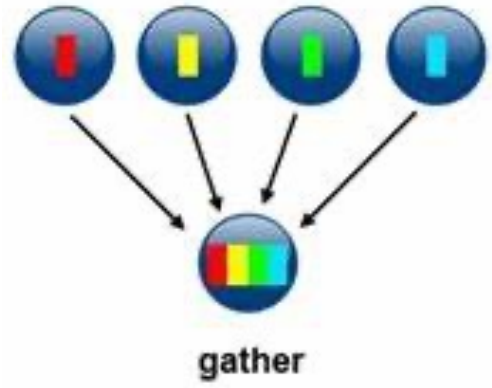
- Paradigma ***Single Program Multiple Data***
  - mesmo binário é executado em vários computadores diferentes, simultaneamente.
  - Processos recebem **parte** do volume total de dados a serem processados:
    - **Paralelismo de dados**: mesma tarefa, mas dados diferentes
    - **Paralelismo de tarefas**: mesmos dados mas tarefas diferentes
  - Quatro das operações providas pelas implementações de MPI:
    - **Broadcast**: ferramenta para espalhar dados
    - **Scatter**: ferramenta para fragmentar dados
    - **Gather**: ferramenta para coletar e compor fragmentos
    - **Reduce**: ferramenta para reduzir resultados parciais

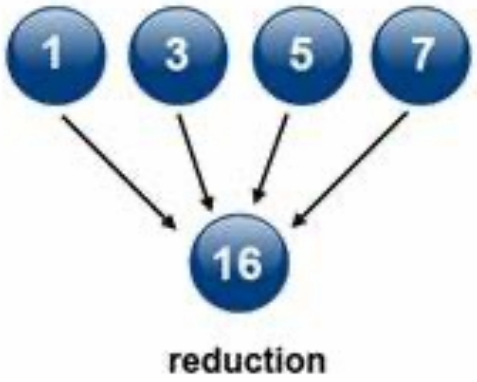
MPI

- Operações básicas:



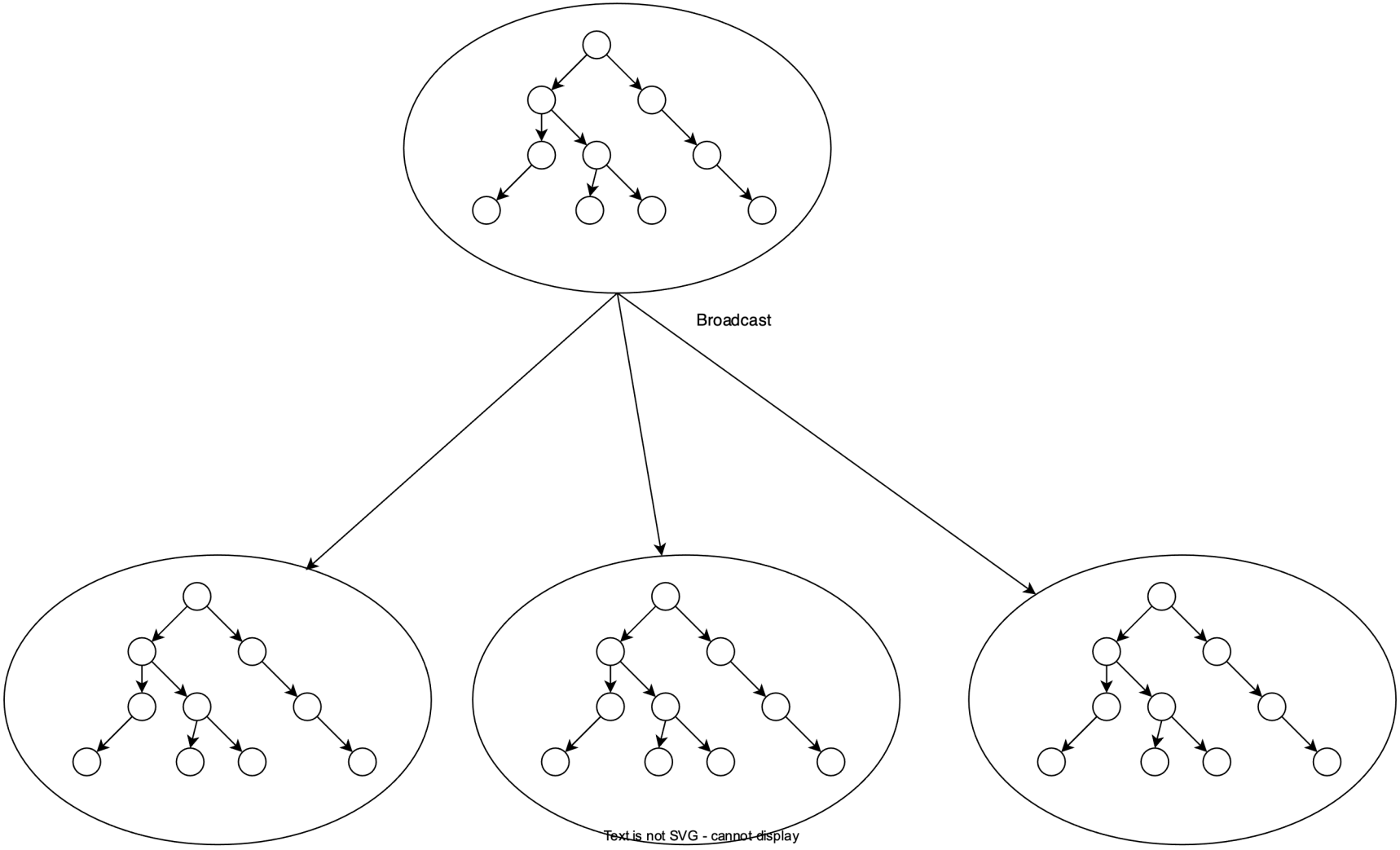






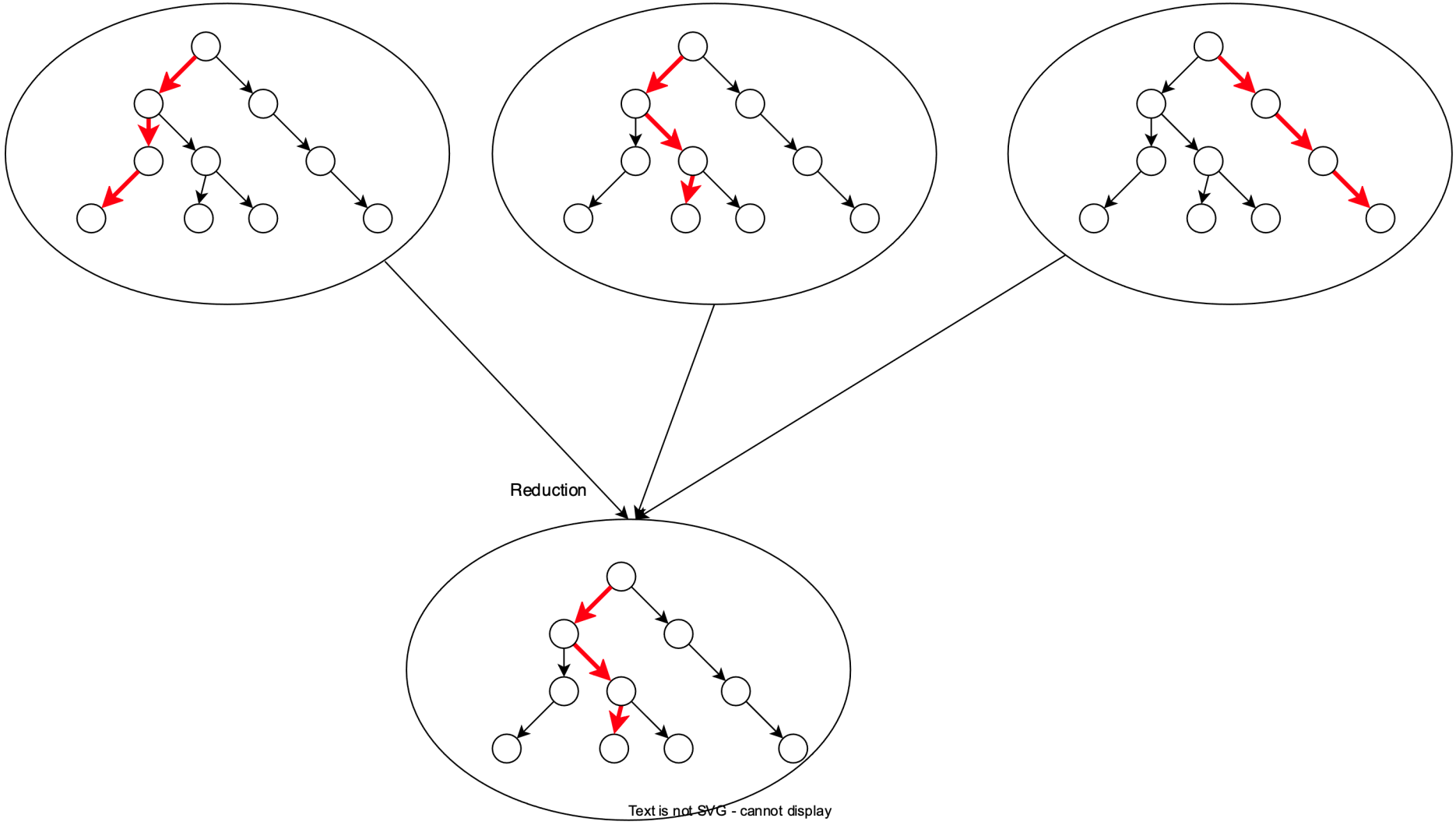
MPI - Exemplo

- Busca em grafo: distribuição com ***broadcast***



MPI - Exemplo

- Busca em grafo: coleta do resultado com ***reduce***

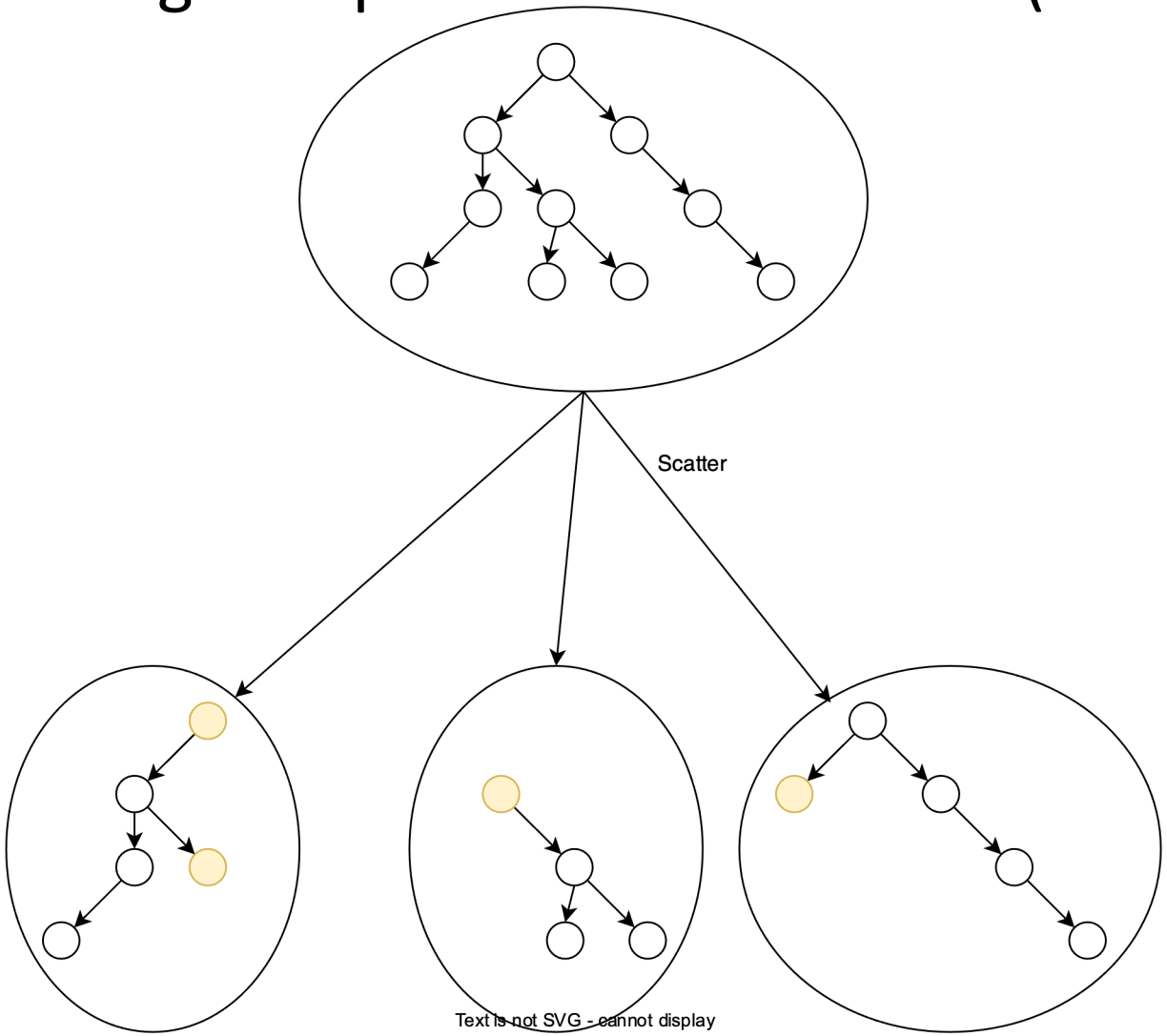


MPI - Exemplo

- Busca em grafo: Abordagem alternativa?

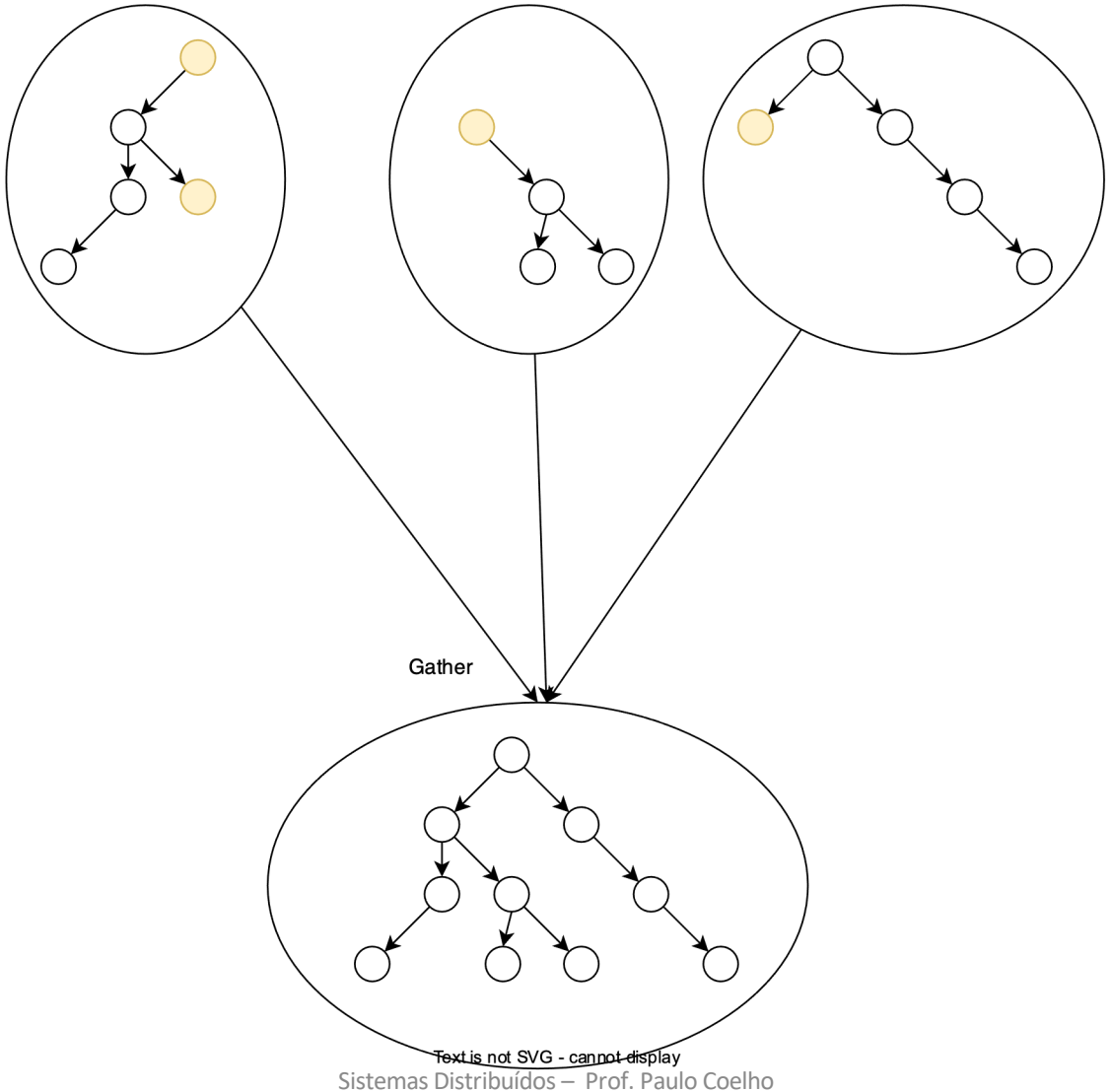
MPI - Exemplo

- Busca em grafo: paralelismo de dados (***scatter***)



MPI - Exemplo

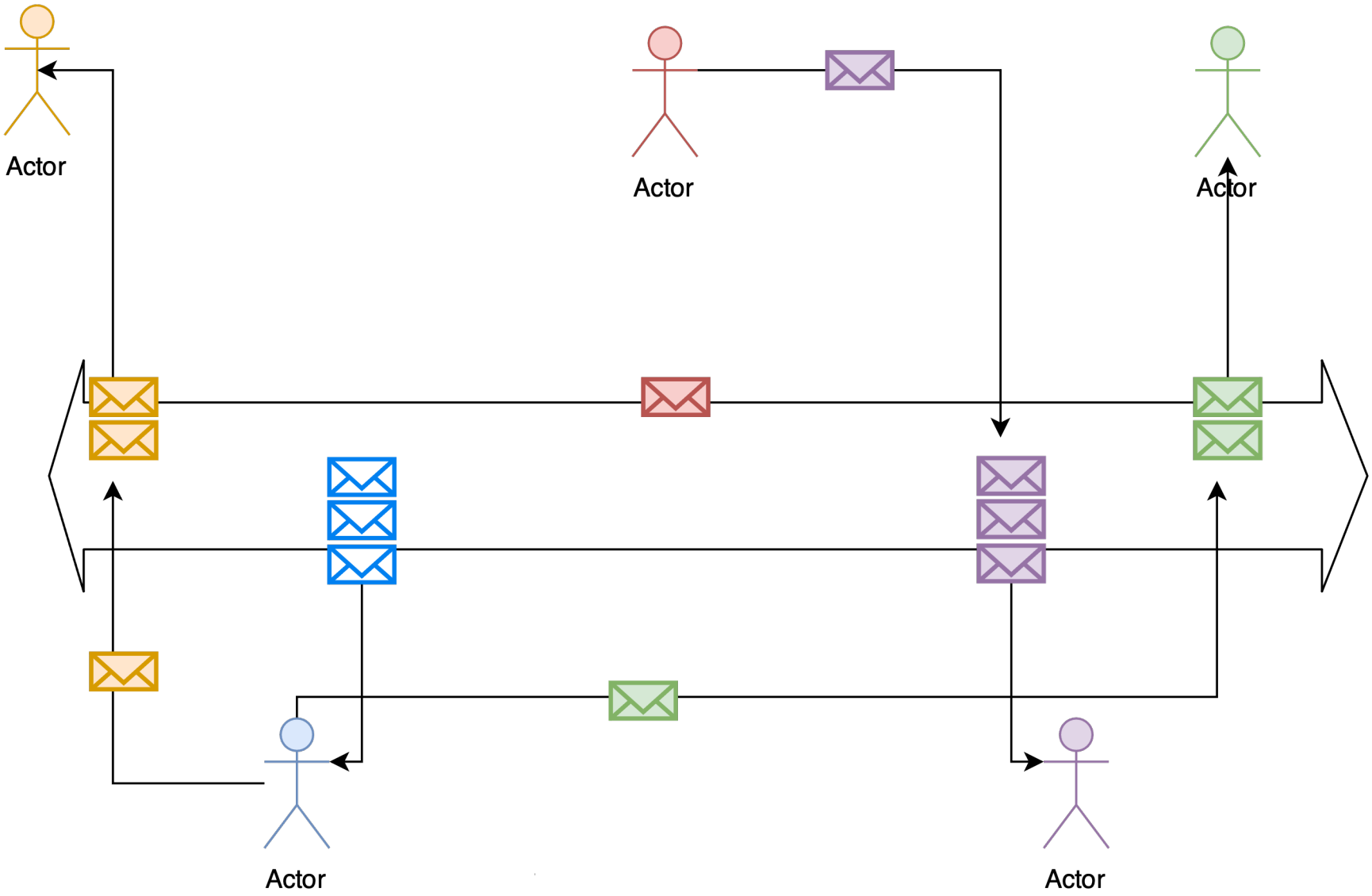
- Busca em grafo: paralelismo de dados (***gather***)



MQ

- Filas de mensagens (*Message Queues – MQ*)
  - Forma de encaminhar dados para nós específicos sem a necessidade de conexão direta
  - Uso de **caixas de entrada**: semelhante a serviço de e-mail / redes sociais para trocas de mensagens
  - Permitem enfrentar uma das dificuldades de se implementar sistemas distribuídos hoje em dia: a saída/entrada constante de componentes
    - **Desacoplamento temporal**
    - ***Brokers*** devem se manter online para permitir a comunicação
  - Notoriedade recente:
    - Expansão de seu uso em sistemas com arquiteturas microsserviços

MQ



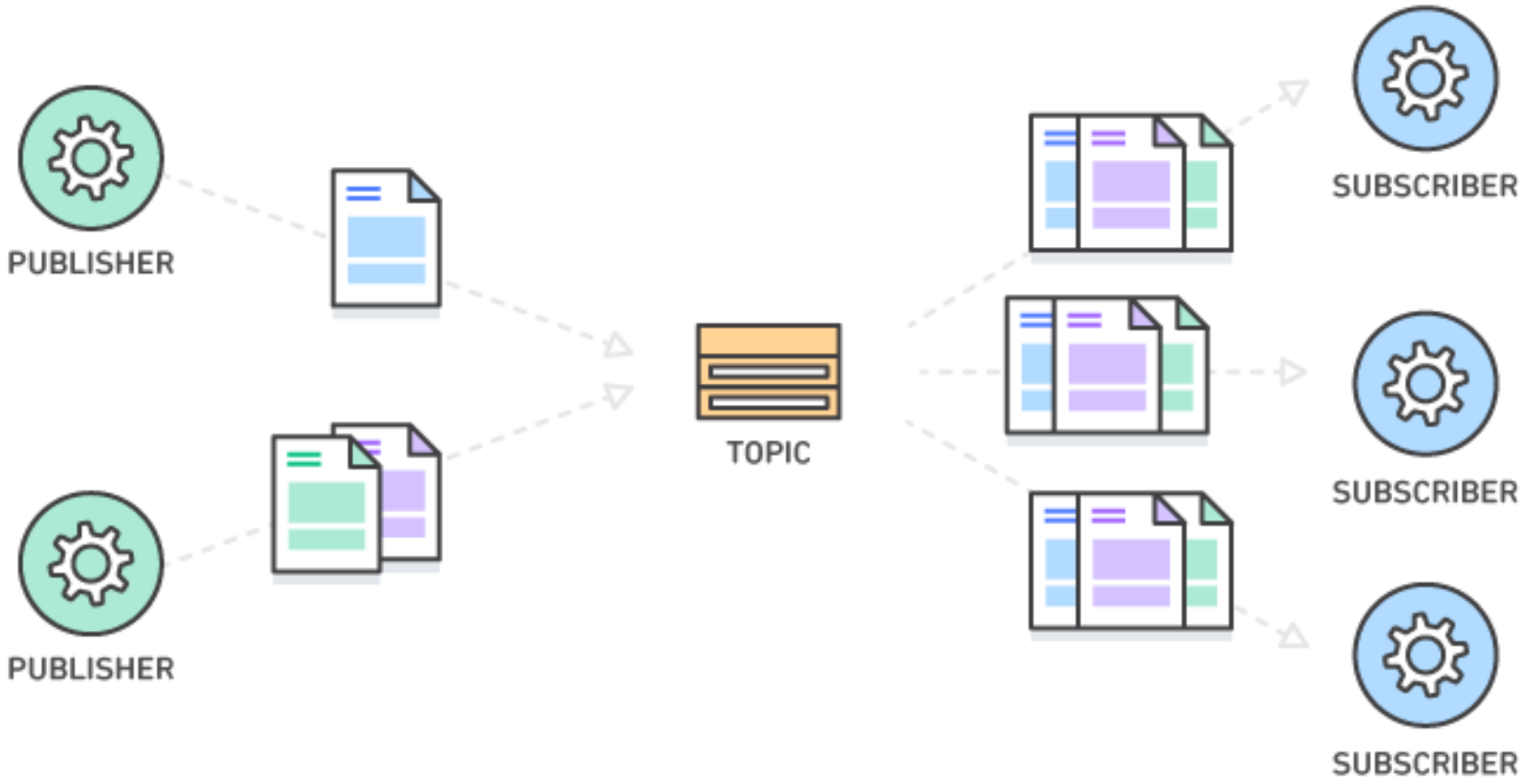
MQ

- Outro problema permanece:
  - Como fazer com que todos se conheçam e que cada um saiba exatamente qual informação deve disponibilizar para cada outro?
    - Contactar individualmente cada um dos usuários da mesma rede para perguntar se está interessado?
  - Outro MOM permite que mensagens sejam ofertadas no sistema distribuído de acordo com assunto (tópico) de interesse:
    - Mensagens apenas são entregues a processos que se declarem interessados no tópico

Publish/Subscribe

- Demais mecanismos exigem que os processos se identifiquem
- Na comunicação *publish/subscribe* (ou *pub/sub*), esta este requisito não está mais presente.
- Processo que envia uma mensagem, ***publisher***, não envia mensagens para um destinatário.
  - Em vez disso, publica mensagens com um **tópico**, aos quais os ***subscribers*** se subscrevem.
- A comunicação não acontece diretamente, mas via *brokers*
- *Publishers* e *subscribers* não precisam executar ao mesmo tempo ou sequer saber da existência um do outro.

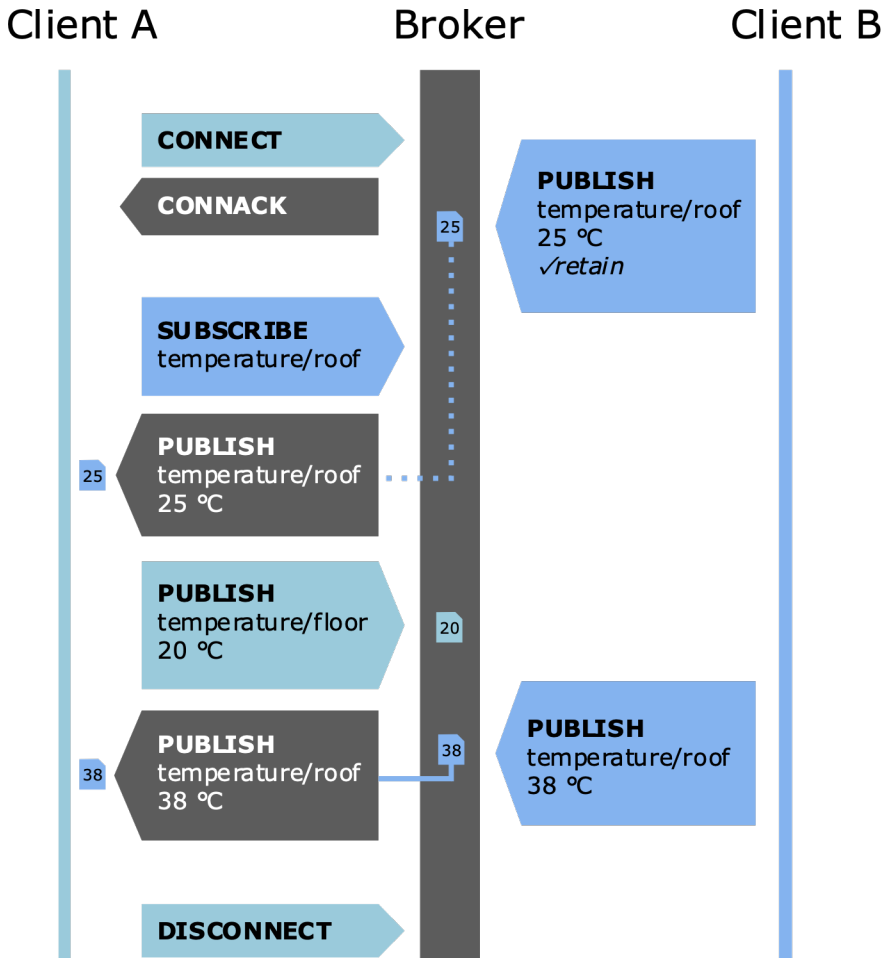
Publish/Subscribe



Publish/Subscribe

- Desacoplamento em várias dimensões das partes envolvidas.
  - **Espaço**: *publishers* e *subscribers* não precisam se conhecer
  - **Tempo**: *publishers* e *subscribers* não precisam estar em execução ao mesmo tempo
  - **Sincronização**: operações não precisam ser interrompida durante a publicação ou recebimento
- *Brokers:*
  - Dão persistência às mensagens (caso necessário)
  - Permitem especificação de filtros de mensagens associados às subscrições:
    - Baseada em **assunto**: *subscribers* se registram para receber mensagens de um ou mais tópicos de interesse. Exemplo: /devices/sensor/+/temperature.
    - Baseada em **conteúdo**: baseada em linguagem de filtragem de conteúdo específica. *Downside:* mensagem não pode ser criptografada.
    - Baseada em **tipo**: leva em consideração o tipo ou classe de uma mensagem ou evento, como o tipo *Exception* e subtipos, por exemplo.
  - Mesma mensagem pode ser entregue a múltiplos *subscribers* se pertencer a um tópico de interesse em comum
  - Mesmo *subscriber* pode se interessar por diversos tópicos

Publish/Subscribe – MQTT



- **MQ** Telemetry **T**ransport
- Mensagem de controle:
  - de 2B a 256MB
- 14 tipos de mensagens:
  - Conexão e desconexão, publicação, confirmação, supervisão, etc.
- MQTT utiliza protocolo TCP
- MQTT-SN (*Sensor Network*) utilizado para outros protocolos:
  - UDP, zigbee, bluetooth.
- Envia credenciais em texto plano
- Segurança pode ser provida com TLS
- Portas:
  - 1883: sem criptografia
  - 8883: com criptografia

Protocolos epidêmicos

- Espelhados no modo como boatos ou doenças se propagam entre um conjunto de indivíduos; este tipo de protocolo é conhecido como ***gossiping*** (fofoca) ou **epidêmico**.
- Resolvem um problema de ***multicast*** de mensagens, isto é, de comunicação um para muitos
- Não há a figura de uma entidade coordenadora
- Entrega de mensagens é garantida probabilisticamente
- Na prática, usados:
  - Na descoberta de nó em um sistema distribuído
  - Para propagar informações sobre seus estados individuais
  - Para computar informações globais

Protocolos Epidêmicos – algoritmo

- Considere o seguinte algoritmo básico, que executa em rodadas.
- **Algoritmo básico:**

```
A cada rodada
   para cada nó p infectado com o vírus v
      escolha F outros processos p_i,
      aleatoriamente
         para cada p_i, contamine p_i com v
```

- **Por que rodadas**
  - Permitir que um conjunto de novos dados seja consolidado para aproveitar cada interação para passar um pacote já consolidado de dados adiante

Protocolos Epidêmicos – algoritmo

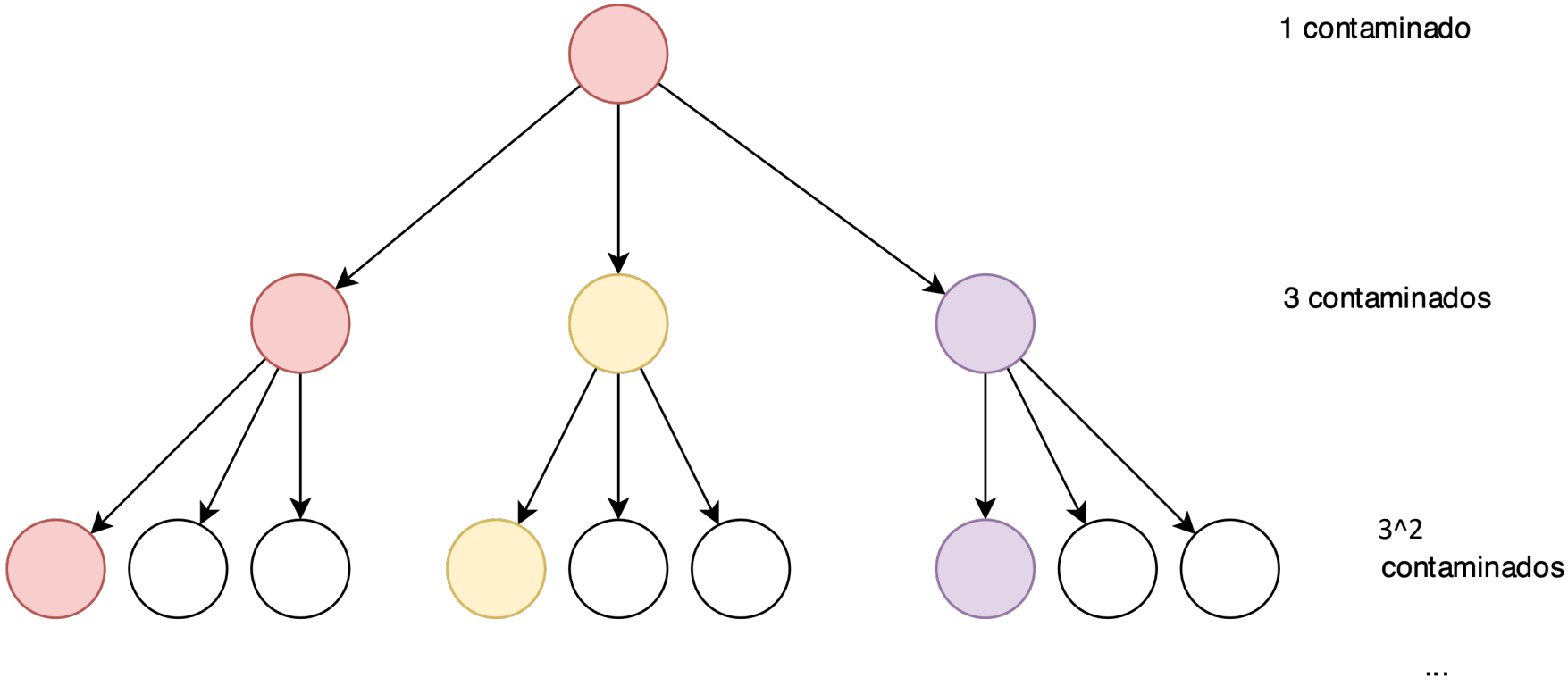
- **Algoritmo básico**

```
A cada rodada
   para cada nó p infectado com o vírus v
      escolha F outros processos p_i, aleatoriamente
         para cada p_i, contamine p_i com v
```

- **Fanout: por que limitar a F?**
  - O objetivo deste tipo de algoritmo é escalar para centenas, milhares ou até milhões de processos
  - É preciso limitar a quantidade de interações a cada ciclo por razões práticas
  - Considere o caso em que F=2:
    - no início do algoritmo, temos 1 processo infectado
    - Durante o primeiro ciclo o número de infectados é 3
    - Assim por diante

Protocolos Epidêmicos – algoritmo

- **Fanout = 2**



Protocolos Epidêmicos – push / pull

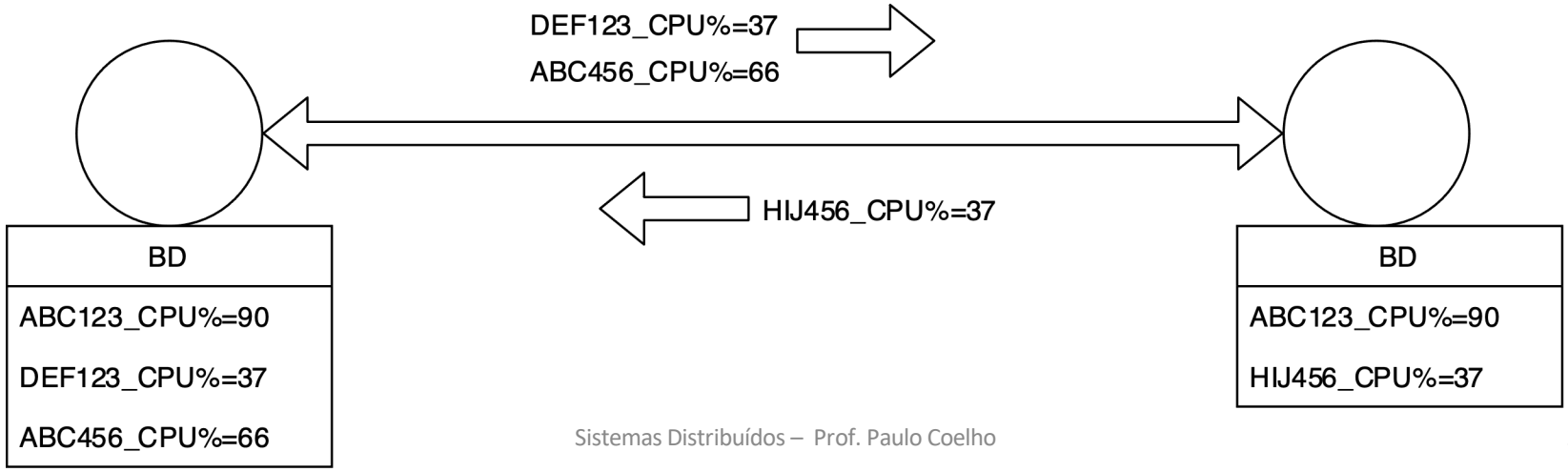
- Nós infectados contatam outros nós para então fazer um ***push*** do vírus
- Para acelerar a propagação, processos saudáveis podem tentar se infectar contactando outros nós e fazendo um ***pull*** dos vírus presentes no outro
- Ou ainda pode-se fazer um misto das duas anteriores: ***push/pull***,
  - Faz mais sentido em um ambiente com múltiplos vírus circulando.

Protocolos Epidêmicos – Múltiplos vírus

- Em um sistema real:
  - Diversas informações distintas devem estar sendo propagadas, o que é equivalente a ter múltiplos vírus circulando na mesma população
  - Exemplo:
    - Cada processo pode gerar periodicamente um resumo da carga de trabalho com a qual está lidando e propagar este resumo para todo o sistema, para ser usado como entrada em uma política de balanceamento de carga.

Protocolos Epidêmicos – Múltiplos vírus

- Modelo comum de representação dos dados em cada nó é um mapa, um banco de dados **chave/valor**, onde a chave é um identificador único da informação e o valor é dado em si, por exemplo ("ABCD1234_CPU%","90%")
- A cada contato, um nó faz um *pull* das entradas cujas chaves desconhece e um *push* das que a contraparte não conhece.



Protocolos Epidêmicos – Aspectos práticos

- **Frescor da informação**
- **Coleta de lixo**:
  - Informações muito antigas devem ser esquecidas em algum momento
  - Necessidade de “lembrar” da exclusão (imunidade):
    - **filtros de bloom**