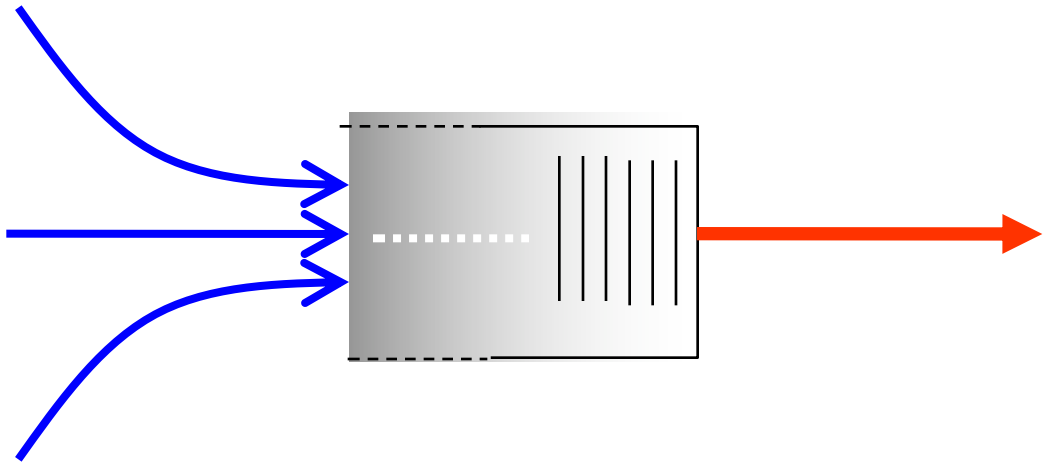
Controle de Congestionamento TCP

Slides by Jennifer Rexford. Used with permission.

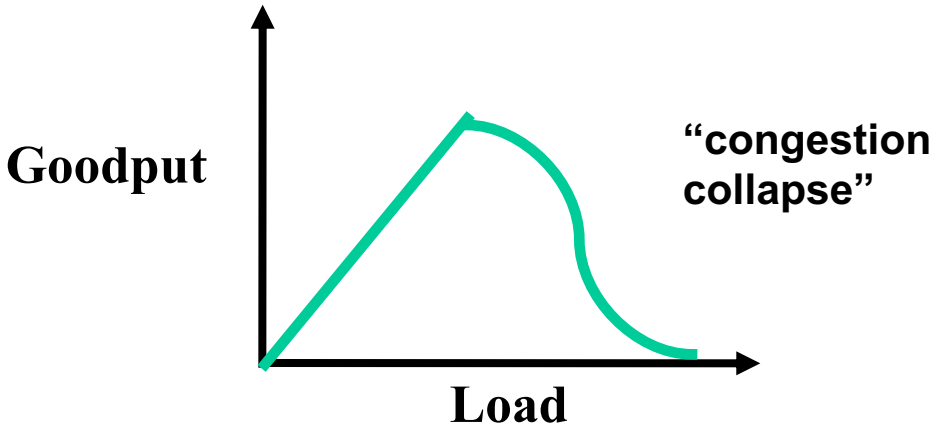
Congestionamento é Inevitável em IP

- Entrega de maior esforço
  - Deixa todo mundo enviar
  - Tenta entregar o que pode
  - … e simplesmente descarta o resto
- Se muitos pacotes chegam em um período pequeno de tempo
  - O nó não pode processar todo o tráfego que chega
  - … e o buffer pode eventualmente transbordar



O Problema de Congestionamento

- O que é congestionamento?
  - Carga é maior do que a capacidade
- O que os roteadores IP fazem?
  - Descartam o excedente de pacotes
- Por que isso é ruim?
  - Desperdício de LB com retransmissões

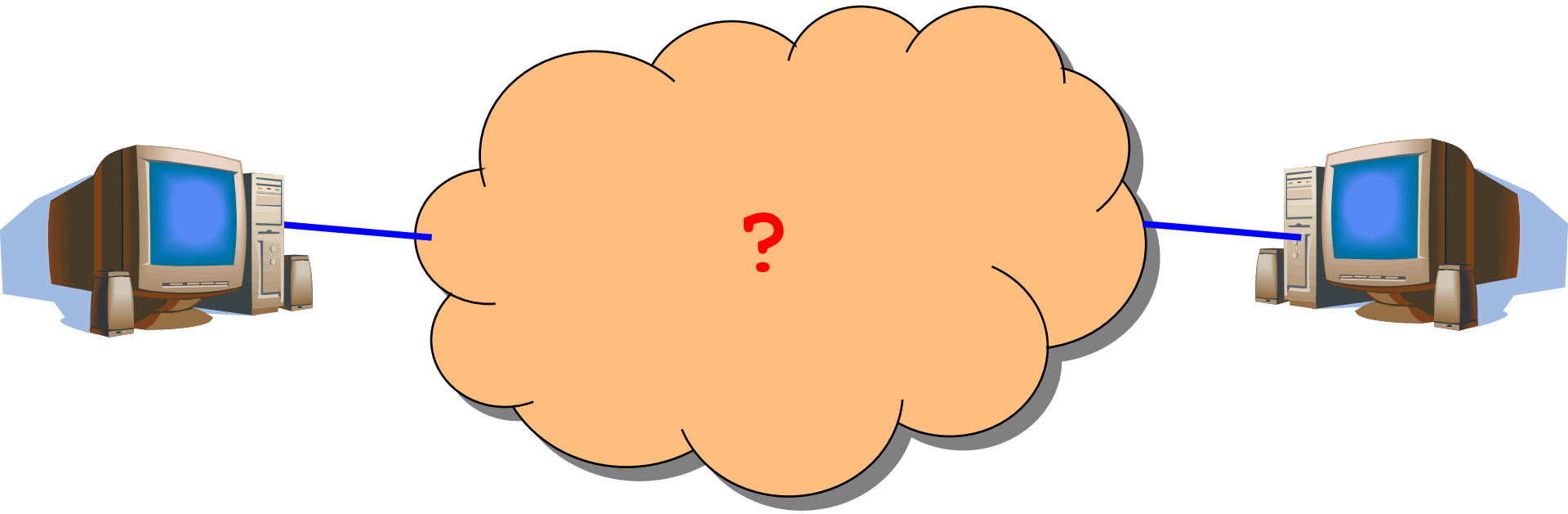


Increase in load that results in a *decrease* in useful work done.

Muitas Questões Importantes

- Como o transmissor sabe se há congestionamento?
  - Feedback explícito da rede?
  - Inferência com base no desempenho da rede?
- Como o transmissor deveria se adaptar?
  - Traxa de transmissão calculada explicitamente pela rede?
  - Host coordena com outros hosts?
  - Host pensa globalmente, mas atua localmente?
- Qual é o objetivo de desempenho?
  - Maximizar goodput, mesmo que alguns usuários sofram mais?
  - Fairness? (Whatever the heck *that* means!)
- Quão rápido uma nova conexão TCP deveria transmistir?

Inferência a Partir de Feedback Implícito

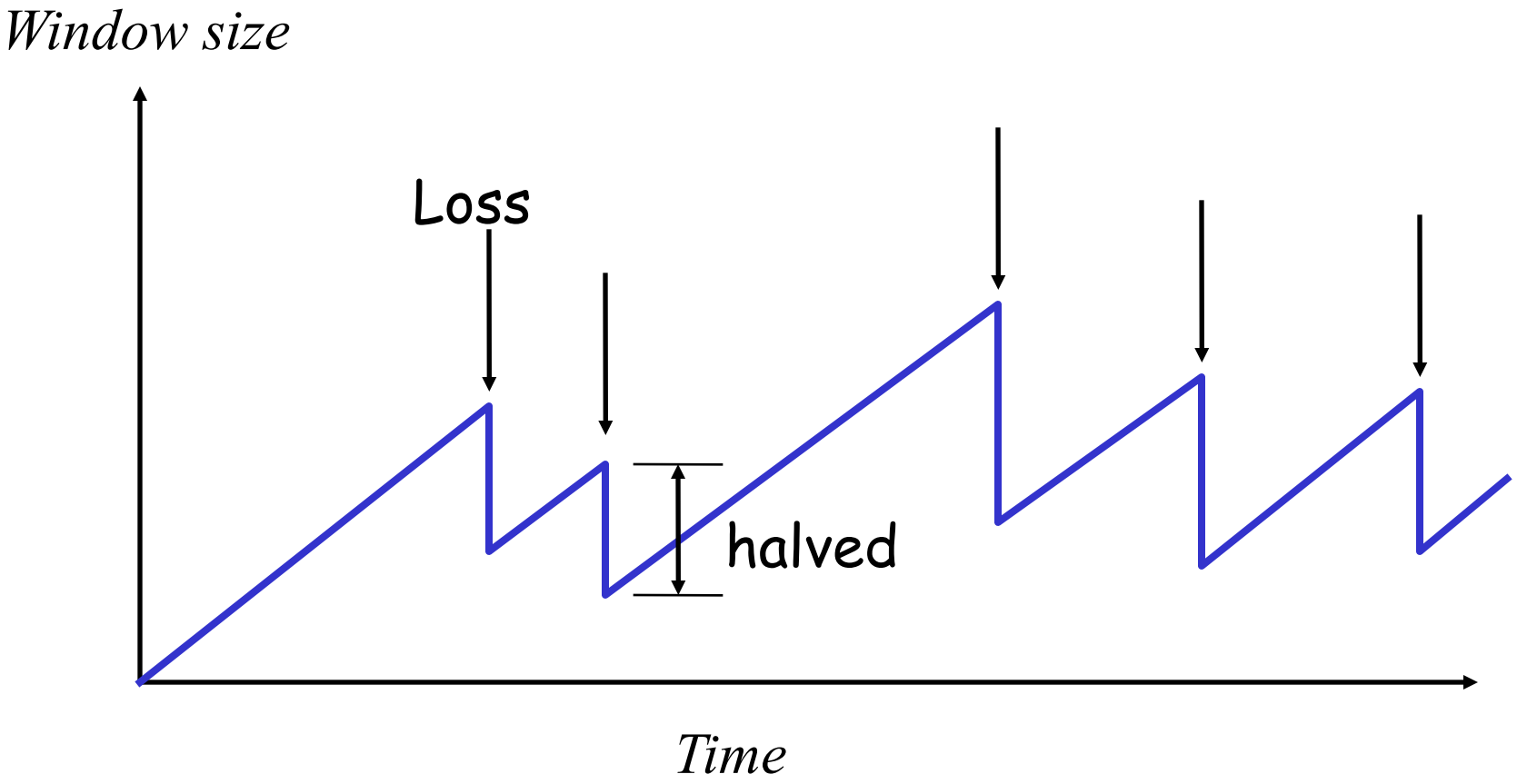


- O que o host enxerga?
  - Perdas em um RTT
  - Atrasos em um RTT

Host Adapta a Taxa de Transmissão ao Longo do Tempo

- Janela de Congestionamento
  - Número máximo de bytes em trânsito
  - i.e., # de bytes ainda esperando confirmação (ACK)
- Ao *detectar* congestionamento
  - *Diminiu* o tamanho da janela (e.g., divide no meio)
  - Host faz a sua parte para aliviar o congestionamento
- Quando *não detecta* congestionamento
  - *Aumenta* o tamanho da janela um pouquinho por vez
  - E observa se os pacotes são entregues com sucesso
  - Host aprende se as condições mudaram

Leva a TCP "Sawtooth"

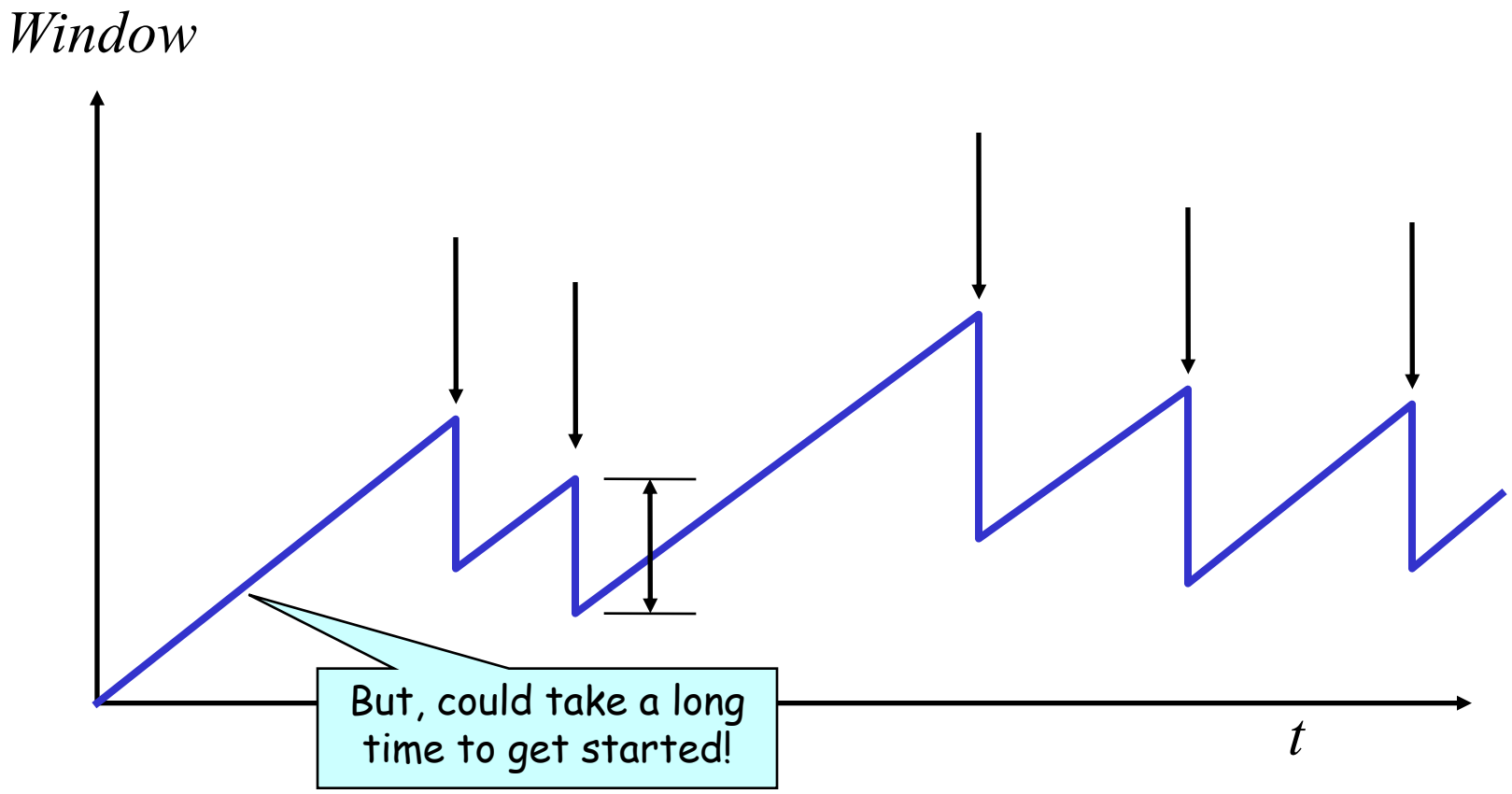


Receiver Window vs. Congestion Window

- Controle de Fluxo
  - Evita que um *transmissor rápido* sobrecarregue um *receptor lento*
- Controle de Congestionamento
  - Evita que um *conjunto de transmissores* sobrecarregue a *rede*
- Conceitos diferentes, mas mecanismos semelhantes
  - Controle de fluxo TCP: receiver window
  - Controle de congestionamento TCP: congestion window
  - TCP window: min{congestion window, receiver window}

Como um Novo Fluxo Deveria Começar

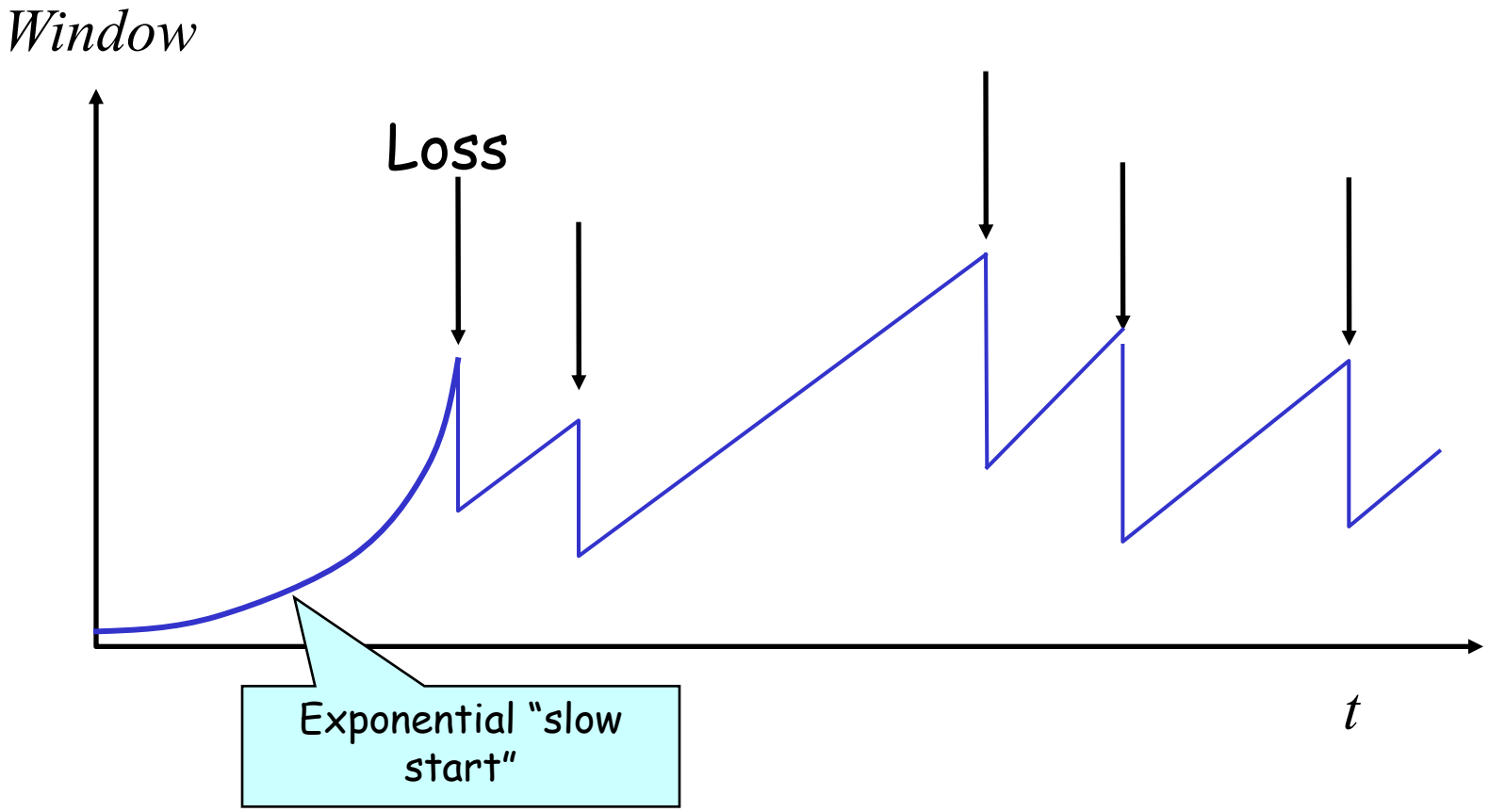
Precisa começar com uma CWND pequena para não sobrecarregar a rede.



Fase "Slow Start"

- Começa com uma janela de congestionamento pequena
  - Inicialmente, CWND é 1 Max Segment Size (MSS)
  - Portanto, a taxa de transmissão inicial é MSS/RTT
- Isso pode ser um grande desperdício
  - Pode ser bem menor do que a capacidade real
  - Crescimento linear leva um longo tempo para acelerar
- Fase Slow Start
  - Transmissor começa a uma taxa lenta (justifica o nome)
  - ... mas aumenta a taxa exponencialmente
  - ... até o primeiro evento de perda

Slow Start e TCP Sawtooth

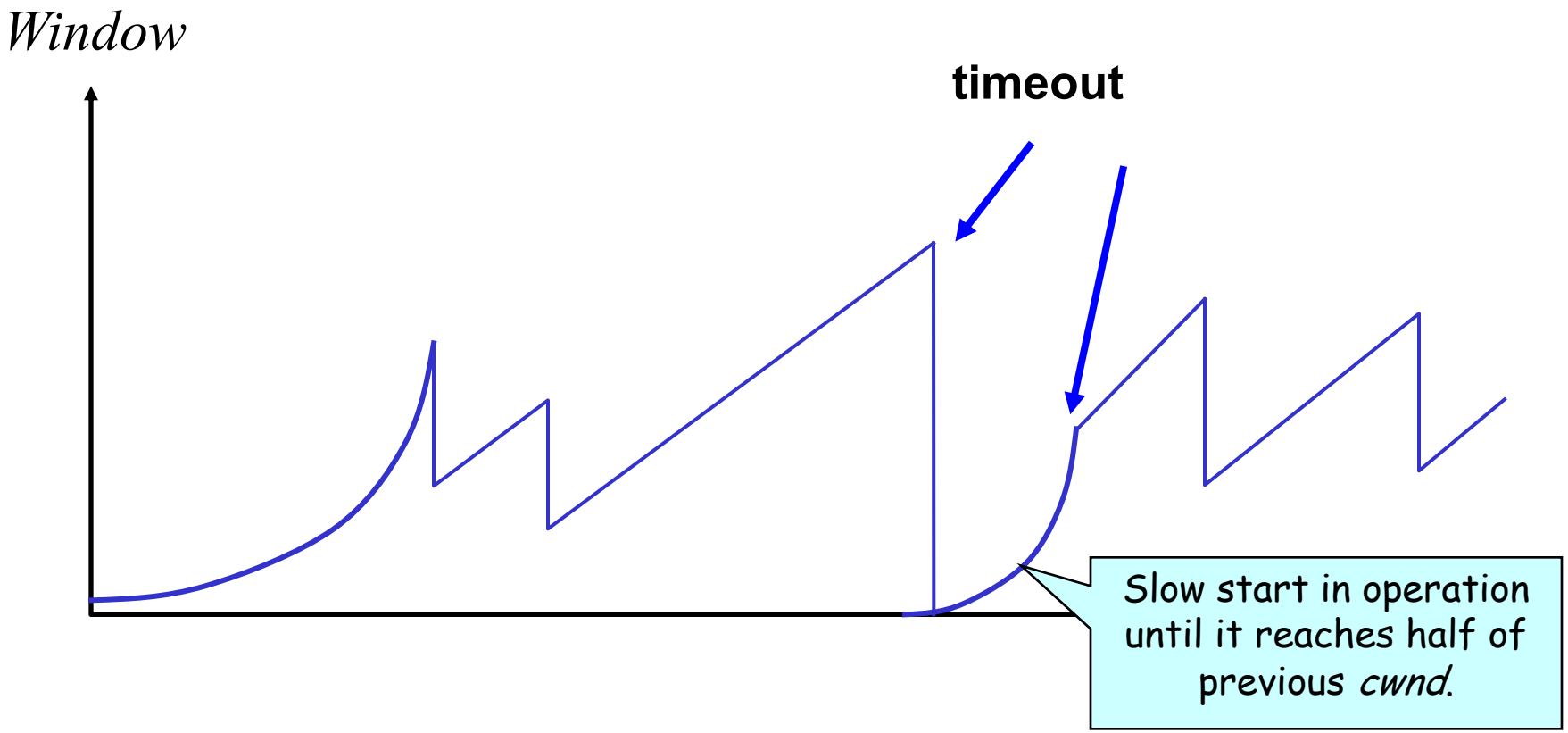


Why is it called slow-start? Because TCP originally had *no congestion control mechanism*. The source would just start by sending a whole receiver window's worth of data.

Dois Tipos de Perda em TCP

- Timeout
  - Pacote n é perdido e detectado por um timeout
    - e.g., porque todos os pacotes em trânsito foram perdidos
  - Depois de timeout, enviar uma CWND inteira poderia disparar uma grande rajada de tráfego
  - Portanto, melhor começar novamente com uma pequena CWND
- Triple duplicate ACK
  - Pacote n é perdido, mas pacotes n+1, n+2, etc. chegam
    - Receptor envia ACKs duplicados
  - E o transmissor retransmiteo pacote n rapidamente
  - Executa um decréscimo multiplicativo e continua

Repetindo Slow Start Depois de Timeout



Slow-start restart: Go back to CWND of 1, but take advantage of knowing the previous value of CWND.

Ineficiências?

- Controle de congestionamento TCP não é muito eficiente
  - O comportamento *sawtooth* leva a desperdícios
  - Fluxos curtos nunca chegam a taxa máxima
  - Baixo desempenho em caminhos com altas LBs
  - Baixo desempenho em caminhos com RTTs longos
- Alguns trabalhos em andamento para melhorar TCP
  - Informações melhores sobre as condições da rede
    - Medições da LB disponível no caminho
    - Feedback explicito dos roteadores
  - Desempenhos melhores sob altos Atraso $\times$ LB (e.g., grandes transmissões de dados entre laboratórios)

Arquitetura de Roteamento da Internet

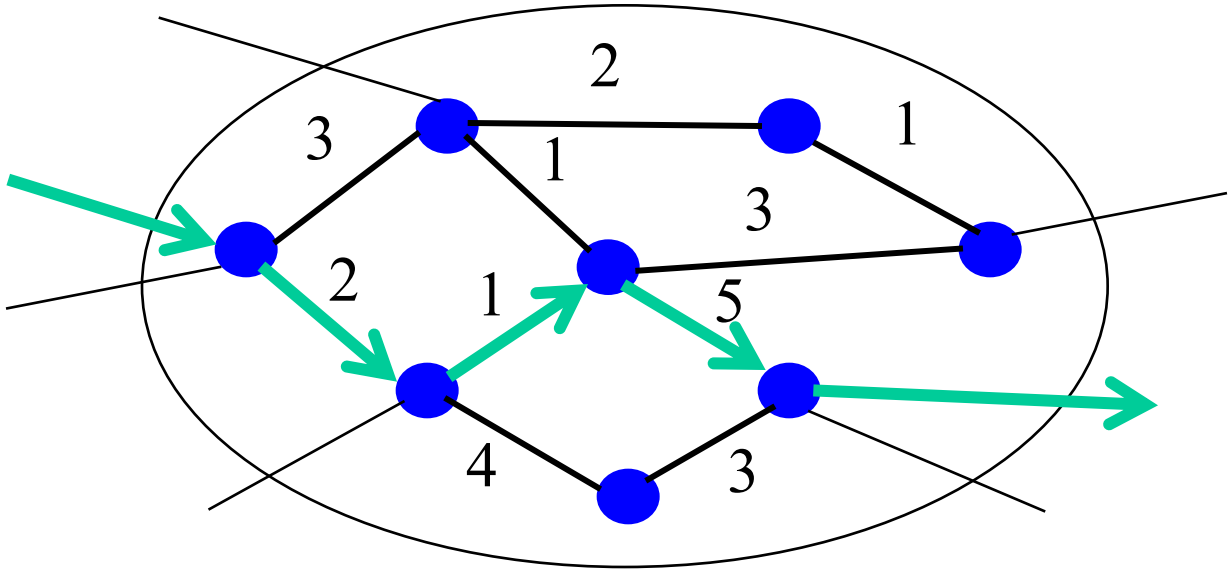
Arquitetura de Roteamento da Internet em Dois Níveis

- Objetivo: gerenciamento distribuído de recursos
  - Interconexão de múltiplas redes
  - Redes sob controles administrativos separados

- Solução: arquitetura de roteamento em dois níveis
  - Intradomínio: dentro de uma região de controle
    - Okay para roteadores compartilhar informações de topologia
    - Roteadores são configurados para atingir um objetivo comum
  - Entre domínios: entre regiões de controle
    - Compartilhamento completo de informações não é okay
    - Redes podem ter objetivos diferentes e conflitantes

Roteamento *Intra*domínio: e.g., Caminho Mais Curto

- Roteadores pertencem à mesma instituição
  - Compartilham um objetivo comum para toda a rede
- Protocolos de roteamento baseados em métricas
  - Tipicamente roteamento de caminho mais curto
  - Com pesos dos enlaces configuráveis



Conclusões

- Arquitetura de roteamento da Internet
  - Two-tiered system
  - Roteamento intradomínio é baseado em métricas, com objetivo comum
  - Roteamento entre domínios é baseado em políticas, precisa conciliar objetivos diferentes entre ASes
- Comportamento do sistema é complexo e misterioso
  - Desafiador para medir
  - Desafiador para caracterizar e diagnosticar

Roteamento Entre Domínios

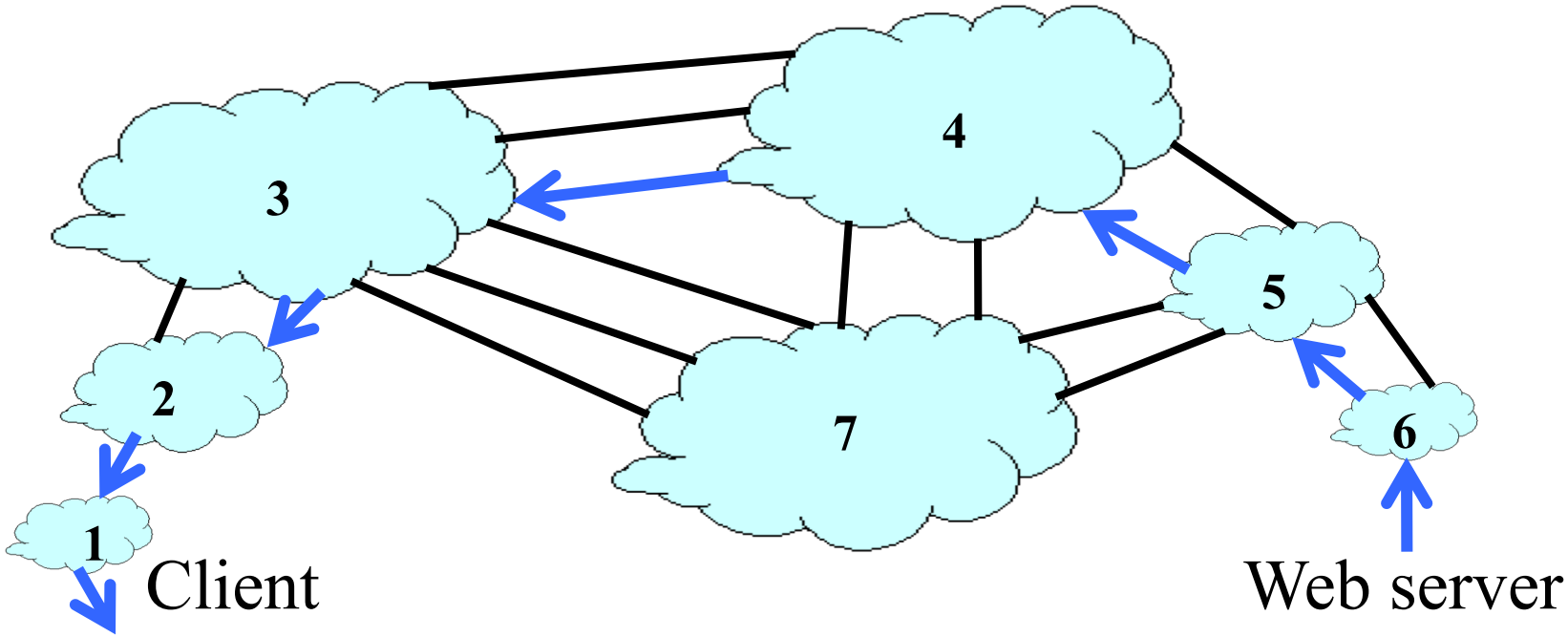
Slides by Jennifer Rexford. Used with permission.

Outline

- Roteamento entre domínios
  - Sistemas Autônomos (*Autonomous Systems* – ASes)
- Roteamento por vetor de caminhos
  - Detecção de loop e convergência
  - Seleção flexível de caminho
- Relações comerciais
  - Customer-provider e peer-peer
  - Hierarquia de tier-1 ASes para stubs

Roteamento *Entre Domínios*: Entre Redes

- Topologia em nível de AS
  - Nós são sistemas autônomos (ASes - Autonomous Systems)
  - Destinos são prefixos (e.g., 12.0.0.0/8)
  - Arestas são enlaces e relações de negócio



AS Numbers (ASNs)

ASNs são valores de 16 ou 32 bits.
64512 a 65535 são “privados”

Atualmente há mais de 60 mil em uso.

- **Level 3: 1**
- **MIT: 3**
- **Harvard: 11**
- **Purdue: 17**
- **Yale: 29**
- **Princeton: 88**
- **AT&T: 7018, 6341, 5074, ...**
- **UUNET: 701, 702, 284, 12199, ...**
- **Sprint: 1239, 1240, 6211, 6242, ...**
- **...**

ASNs represent units of routing policy

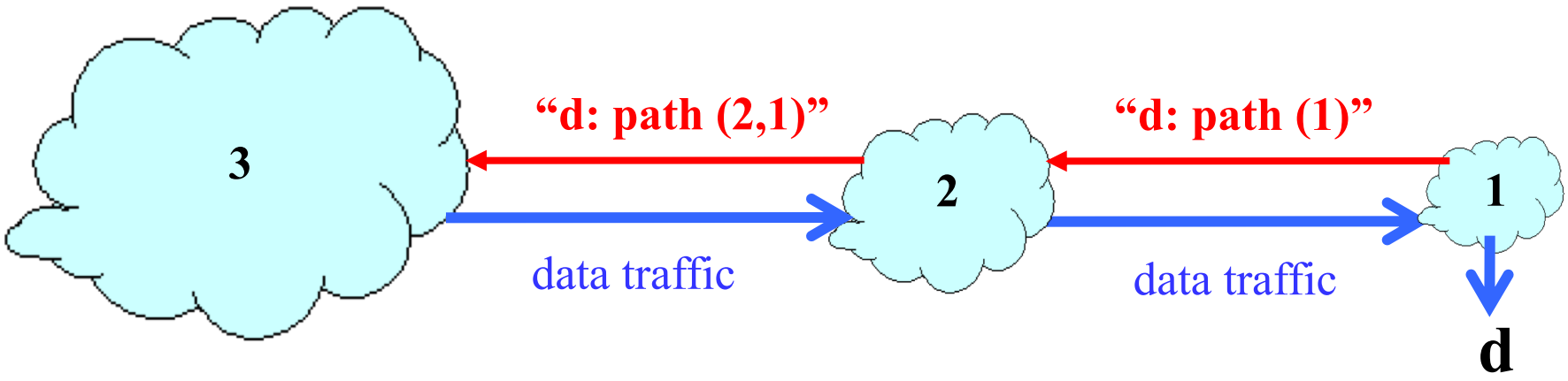
Desafios para Roteamento Entre Domínios

- Escala
  - Prefixos: ~700.000, e crescendo
  - ASes: ~60.000, e crescendo
  - Roteadores: pelo menos na casa dos milhões…
- Privacidade
  - ASes não querem divulgar topologias internas
  - … ou suas relações comerciais com vizinhos
- Política
  - Não há uma noção global de uma métrica de custo de enlace
  - Precisa controlar para onde enviar tráfego
  - … e quem pode enviar tráfego através de você

Roteamento por Vetor de Caminhos e Baseado em Políticas

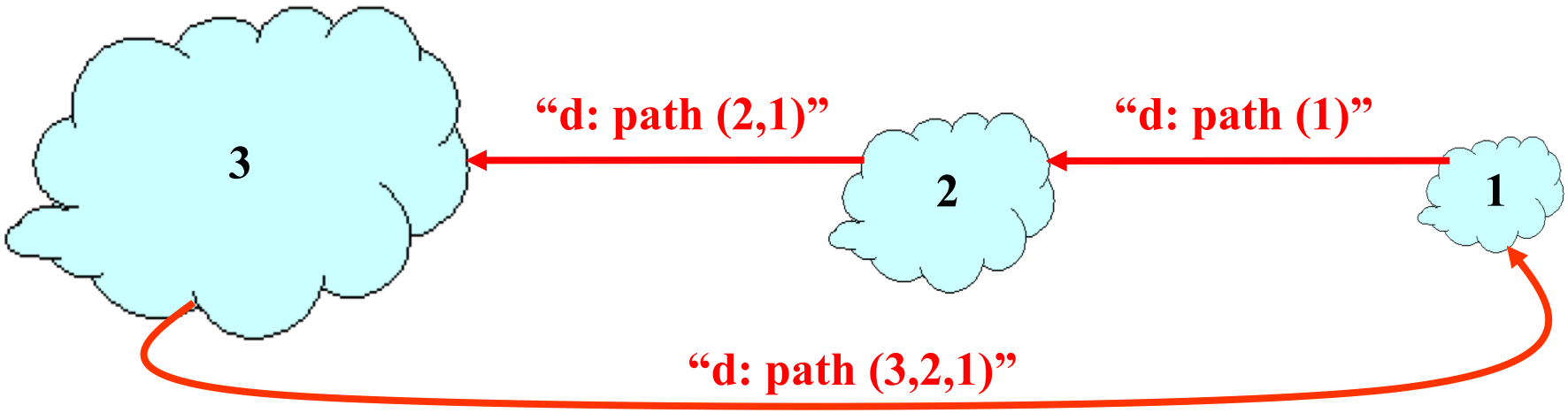
Roteamento por Vetor de Caminhos

- Extensão do roteamento de vetor de distâncias
  - Suporta políticas de roteamento flexíveis
  - Reduz tempo de convergência (evita count-to-infinity)
- Ideia chave: anuncia o caminho inteiro
  - Vetor de distâncias: envia *métrica de distância* por destino d
  - Vetor de caminhos: envia o *caminho inteiro* para destino d



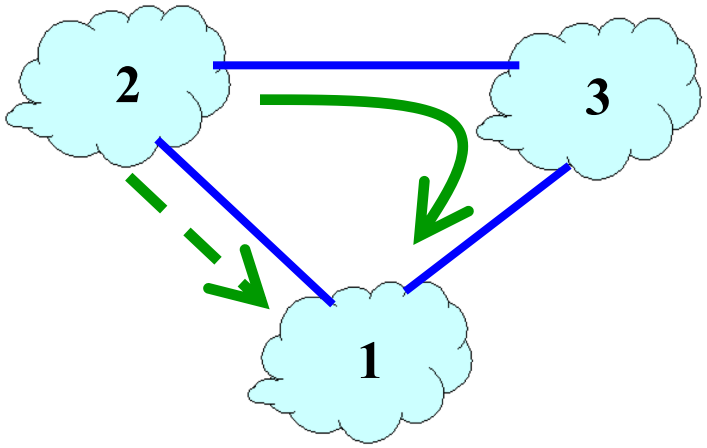
Detecção Mais Rápida de Loop

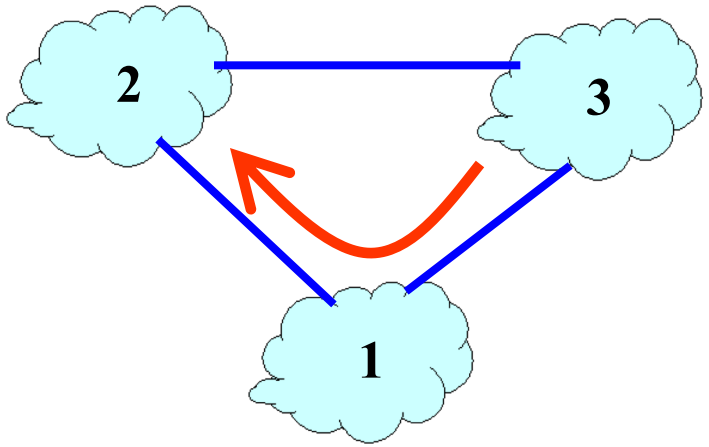
- Nó pode detectar facilmente um loop
  - Verifica se seu próprio identificador está no caminho
  - E.g., nó 1 verifica que está no caminho "3, 2, 1"
- Nó pode simplesmente descartar caminhos com loops
  - E.g., nó 1 simplesmente descarta o anúncio



Políticas Flexíveis

- Cada nó pode aplicar políticas locais
  - Seleção de caminho: Qual caminho utilizar?
  - Exportação de caminho: Anuncia ou não o caminho?
- Exemplos
  - Nó 2 pode preferir o caminho "2, 3, 1" ao invés de "2, 1"
  - Nó 1 pode não anunciar para o nó 3 o caminho "1, 2"





Relações de Negócio

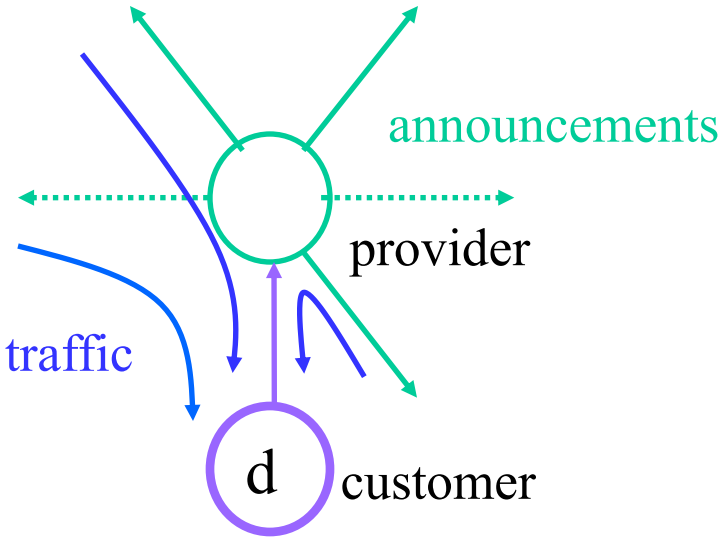
Relações de Negócio

- ASes vizinhos possuem contratos comerciais
  - Quanto tráfego transportar
  - Quais destinos alcançar
  - Quanto pagar em dinheiro
- Relações de negócio comuns
  - Customer-provider
    - E.g., Princeton is a customer of USLEC
    - E.g., MIT is a customer of Level3
  - Peer-peer
    - E.g., UUNET is a peer of Sprint
    - E.g., Harvard is a peer of Harvard Business School

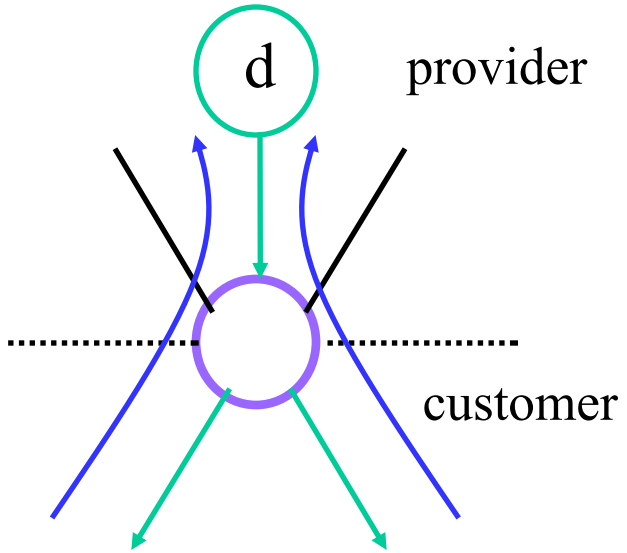
Relacionamento Customer-Provider

- Cliente precisa ser alcançável por todos
  - Provedor conta para todos seus vizinhos como o cliente pode ser alcançado
- Cliente não quer oferecer serviço de trânsito
  - Cliente não deixa seus provedores rotear tráfego por ele

Traffic **to** the customer

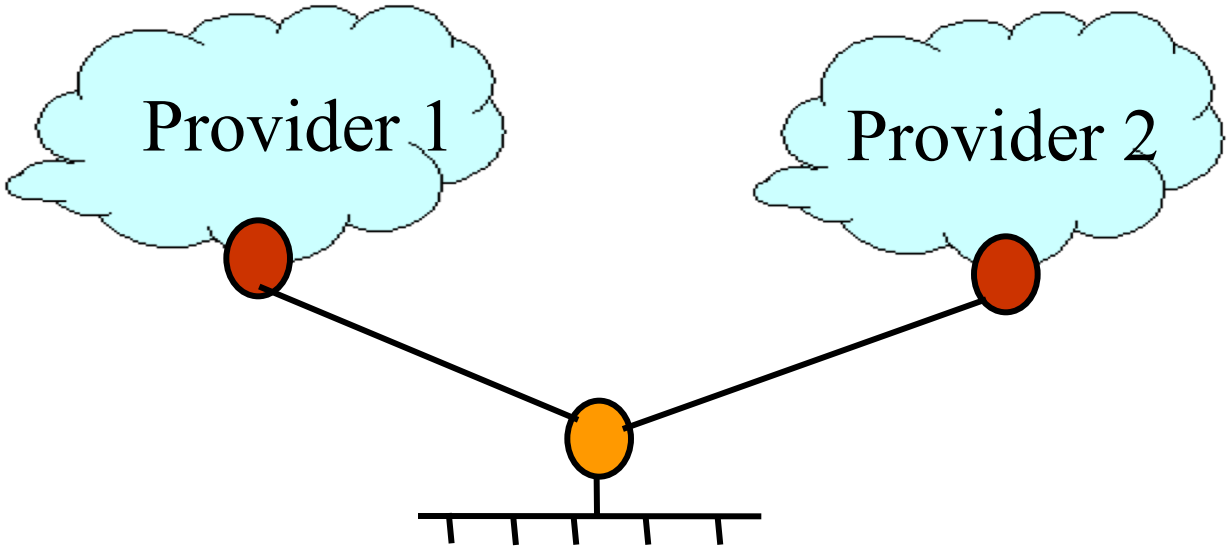


Traffic **from** the customer



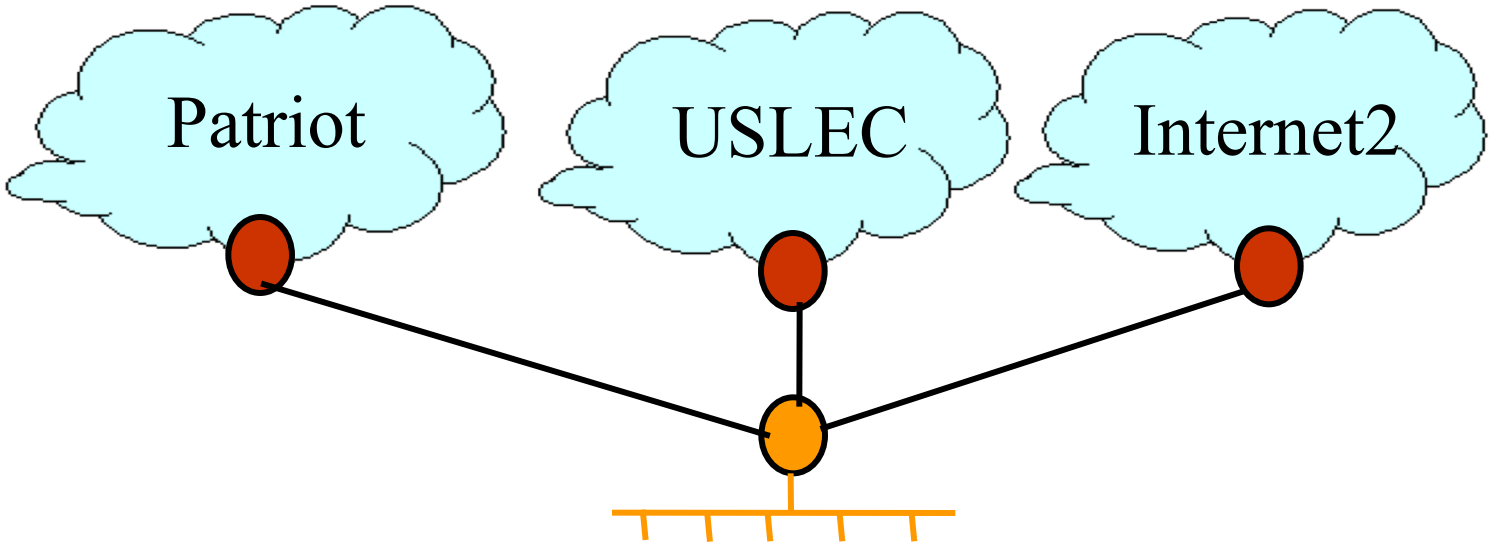
Multi-Homing: Dois ou Mais Provedores

- Motivações para multi-homing
  - Confiabilidade extra, sobrevive a falhas de um único ISP
  - Vantagem financeira via competição
  - Melhor desempenho ao selecionar caminhos melhores



Exemplo de Princeton

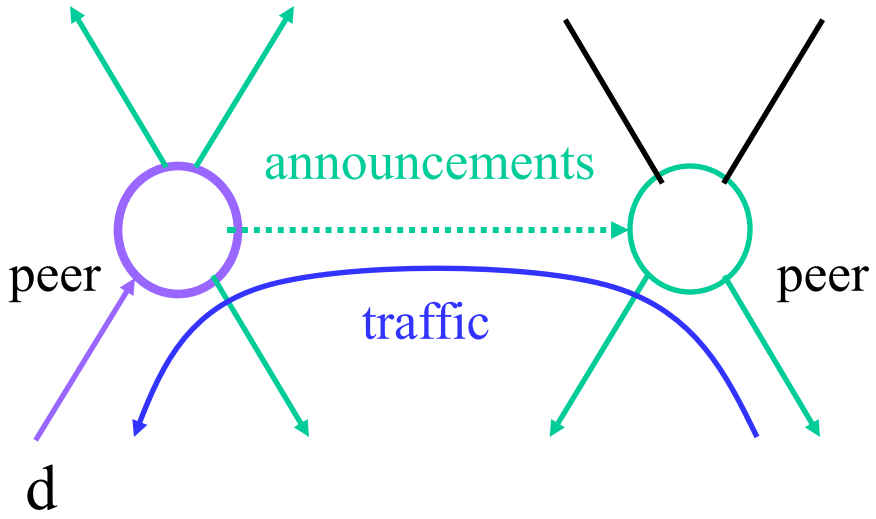
- Internet: cliente de USLEC e Patriot
- Research universities/labs: cliente de Internet2
- Local non-profits: provedor para várias organizações sem fins lucrativos



Relacionamento Peer-Peer

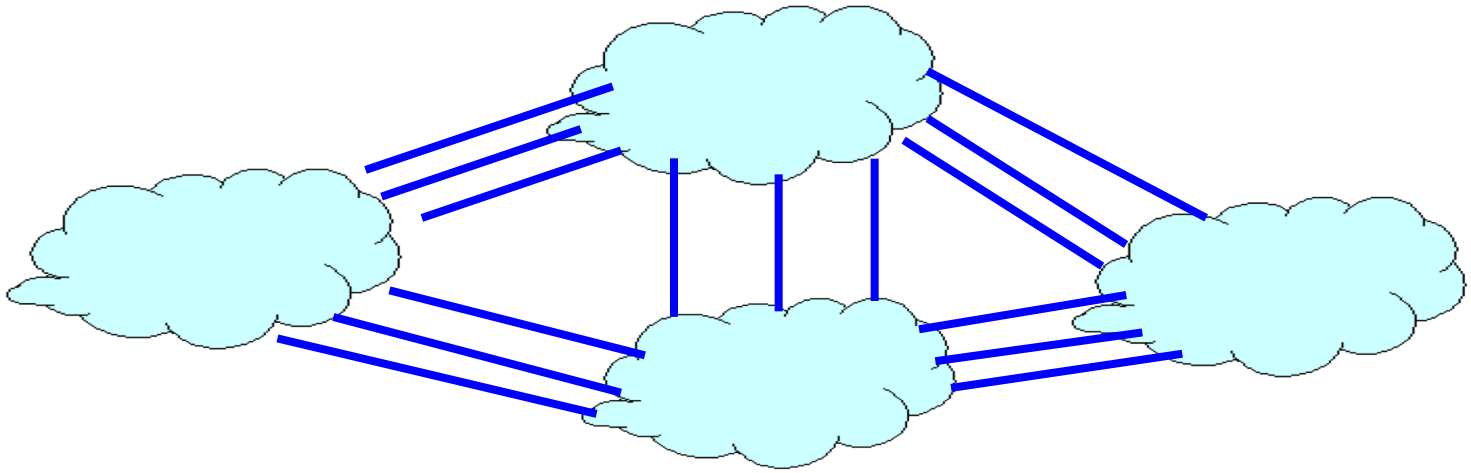
- Peers trocam tráfego entre clientes
  - AS exporta *apenas* rotas dos clientes para um peer
  - AS exporta rotas de um peer *apenas* para seus clientes
  - Geralmente o relacionamento é *settlement-free* (i.e., no $$$)

Traffic to/from the peer and its customers



Estrutura de AS: Provedores Tier-1

- Provedor Tier-1
  - Não possui um provedor *upstream*
  - Geralmente possui um backbone nacional ou internacional
- Topo da hierarquia da Internet ~10 ASes
  - AOL, AT&T, Global Crossing, Level3, UUNET, NTT, Qwest, SAVVIS (formerly Cable & Wireless), and Sprint
  - Conexões peer-peer completas entre provedores tier-1



Estrutura de AS: Outros ASes

- Outros provedores
  - Oferecem serviço de trânsito para clientes *downstream*
  - … mas, precisam de pelo menos um provedor para eles mesmos
  - Geralmente possuem escopo nacional ou regional
  - Representam milhares de ASes.
- ASes Stub
  - Não oferecem serviço de trânsito para outros
  - Conectam-se a um ou mais provedores *upstream*
  - Representam a maioria (e.g., 85-90%) dos ASes

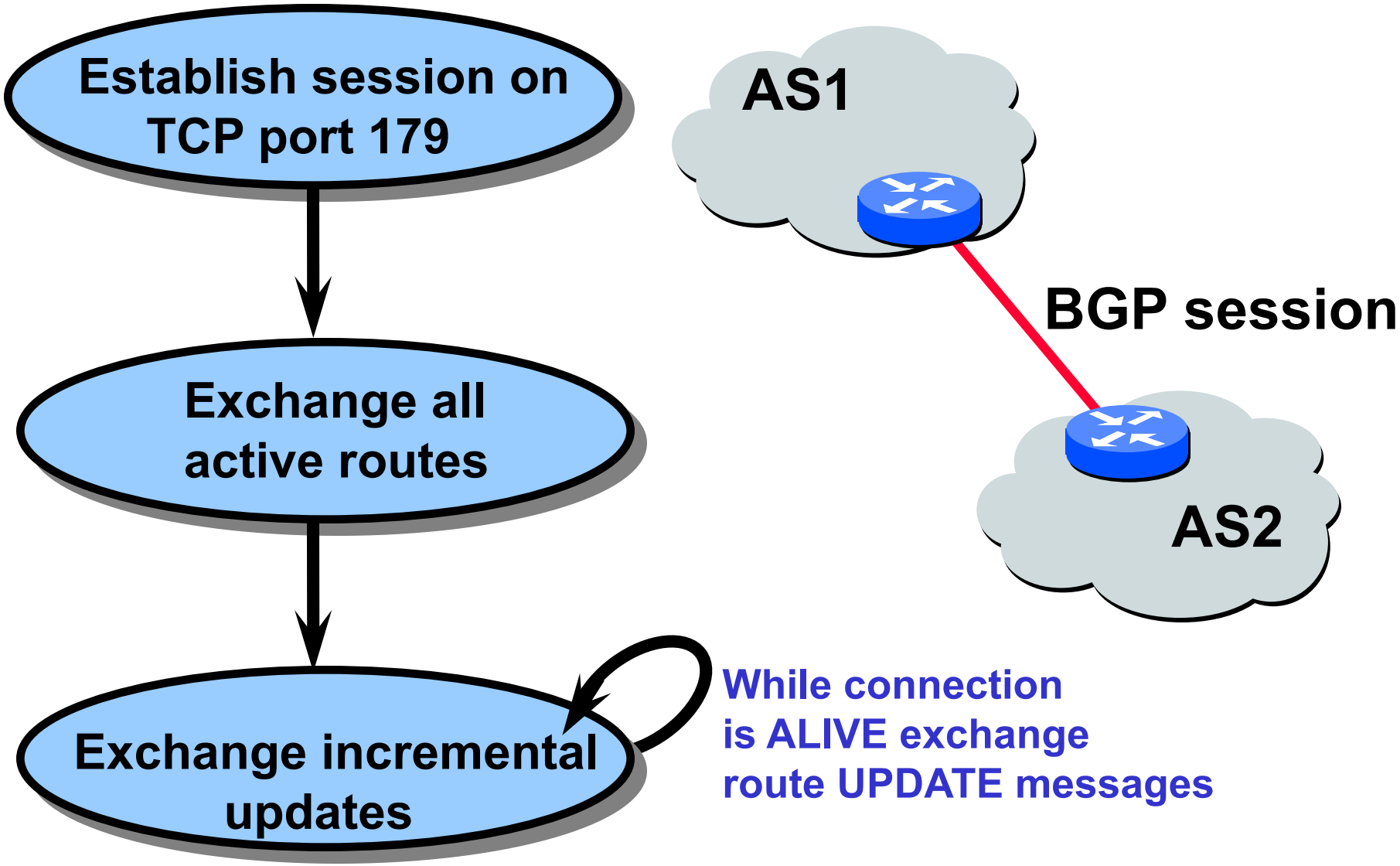
Border Gateway Protocol

Border Gateway Protocol

- Protocolo de vetor de caminhos baseado em prefixos
- Roteamento por políticas baseadas em caminhos de AS
- Evoluiu muito nos últimos anos

- **1989 : BGP-1 [RFC 1105], replacement for EGP**
- **1990 : BGP-2 [RFC 1163]**
- **1991 : BGP-3 [RFC 1267]**
- **1995 : BGP-4 [RFC 1771], support for CIDR**
- **2006 : BGP-4 [RFC 4271], update**

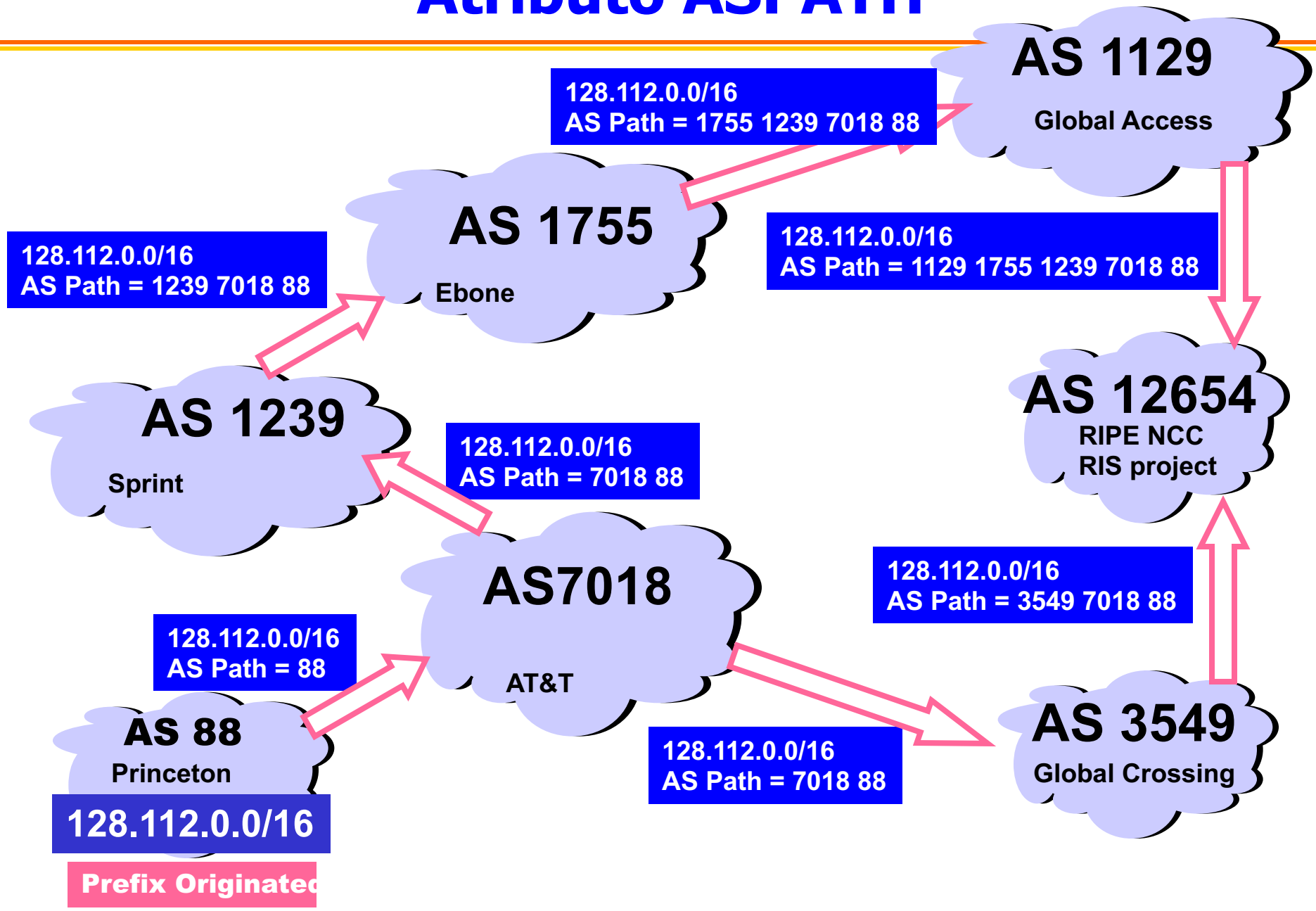
Operações BGP



Protocolo Incremental

- Um nó aprende múltiplos caminhos para um destino
  - Armazena todas as rotas em uma tabela de roteamento
  - Aplica política para selecionar uma única rota ativa
  - … e pode anunciar a rota para seus vizinhos
- Atualizações incrementais
  - Anúncio (*Announcement*)
    - Ao selecionar uma nova rota ativa, adiciona ID do nó (ASN) ao caminho
    - … e (opcionalmente) anuncia para cada vizinho
  - Revogação (*Withdrawal*)
    - Se a rota ativa não está mais disponível
    - … envia uma mensagem de revogação para os vizinhos

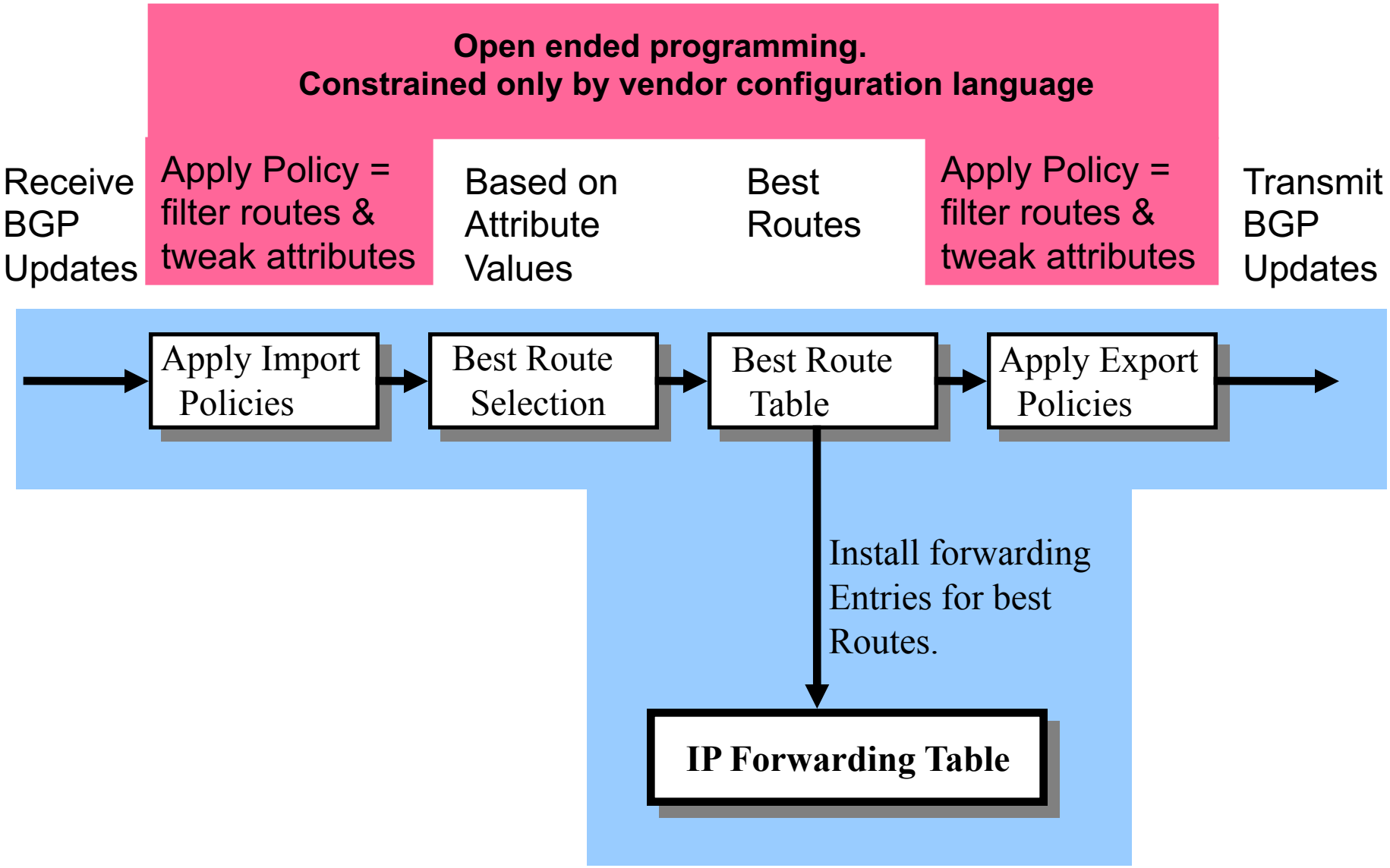
Atributo ASPATH



Política BGP: Aplicando Políticas a Rotas

- Política de Importação (*Import policy*)
  - Filtra rotas indesejáveis do vizinho
    - E.g. prefixo que o cliente não é dono
  - Manipula atributos para influenciar seleção de caminho
    - E.g., atribui preferência local para rotas preferidas
- Política de Exportação (*Export policy*)
  - Filtra rotas que você não quer contar ao vizinho
    - E.g., não conta a um peer uma rota aprendida de outro peer
  - Manipula atributos para controlar o que os outros enxergam
    - E.g., fazer um caminho parecer mais longo do que realmente é

Política BGP: Influenciando Decisões



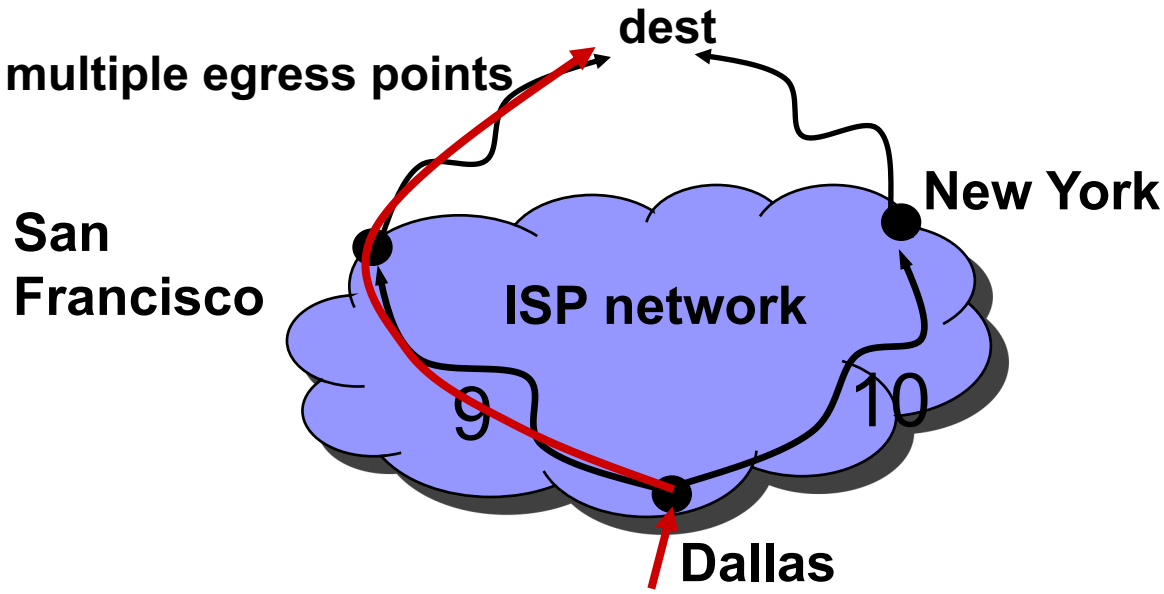
Processo de Decisão BGP em um Roteador

- Routing Information Base (RIB)
  - Armazena todas as rotas BGP para cada prefix de destino
  - Withdrawal message: remove uma rota
  - Advertisement message: atualiza uma rota
- Selecionando a melhor rota
  - Considera todas as rotas BGP para o prefixo
  - Aplica regras para comparar as rotas
  - Seleciona a melhor rota (uma única rota)
    - Usa essa rota na tabela de encaminhamento
    - Envia essa rota para os vizinhos

Processo de Decisão BGP

- Highest local preference
  - *Set by import policies* upon receiving advertisement
- Shortest AS path
  - Included in the route advertisement
- Lowest origin type
  - Included in advertisement or reset by import policy
- Smallest multiple exit discriminator
  - Included in the advertisement or reset by import policy
- Smallest internal path cost to the next hop
  - Based on intradomain routing protocol (e.g., OSPF)
- Smallest next-hop router id
  - Final tie-break

Roteamento Hot-Potato



Hot-potato routing = route to closest egress point when there is more than one BGP route to destination

Causas de Mudanças no Roteamento BGP

- Mudanças na topologia
  - Equipamentos ligando ou desligando
  - Implantação de novos roteadores ou sessões
- Falhas na sessão BGP
  - Devido a falhas de equipamento, manutenção, etc.
  - Ou, devido a congestionamento no caminho físico
- Mudanças na política de roteamento
  - Reconfiguração de preferências
  - Reconfiguração de filtros de rota
- Oscilações persistentes no protoclo
  - Conflitos nas políticas