

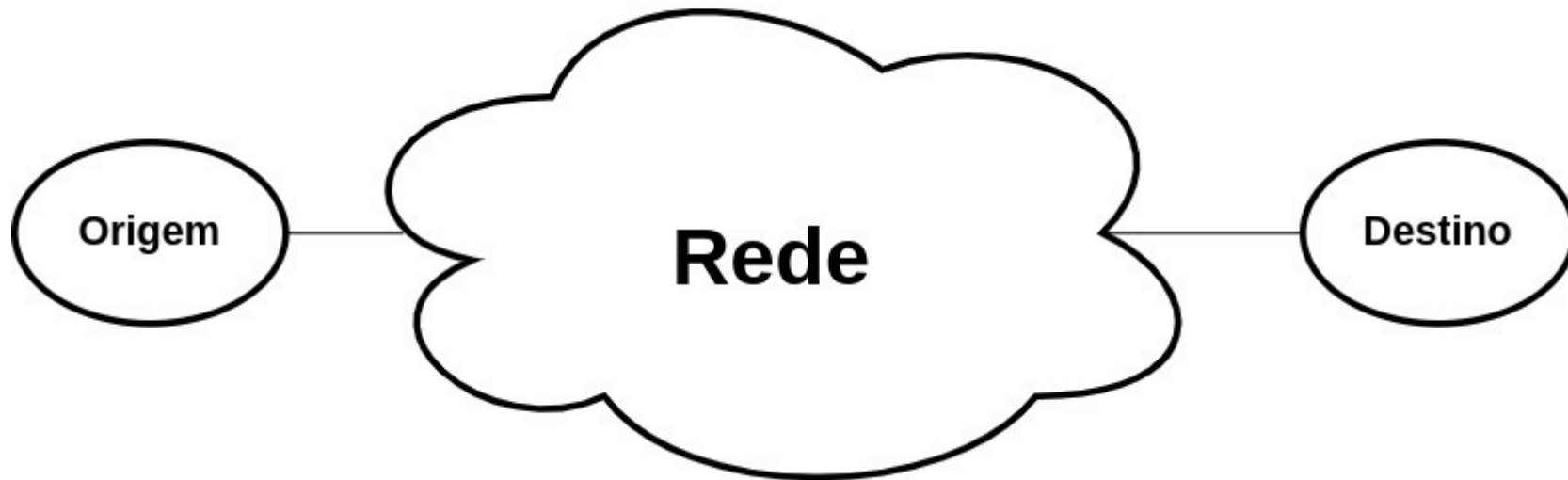
OBJETIVOS

- Hoje continuamos nosso estudo do protocolo TCP
- Foco no Controle de Congestionamento



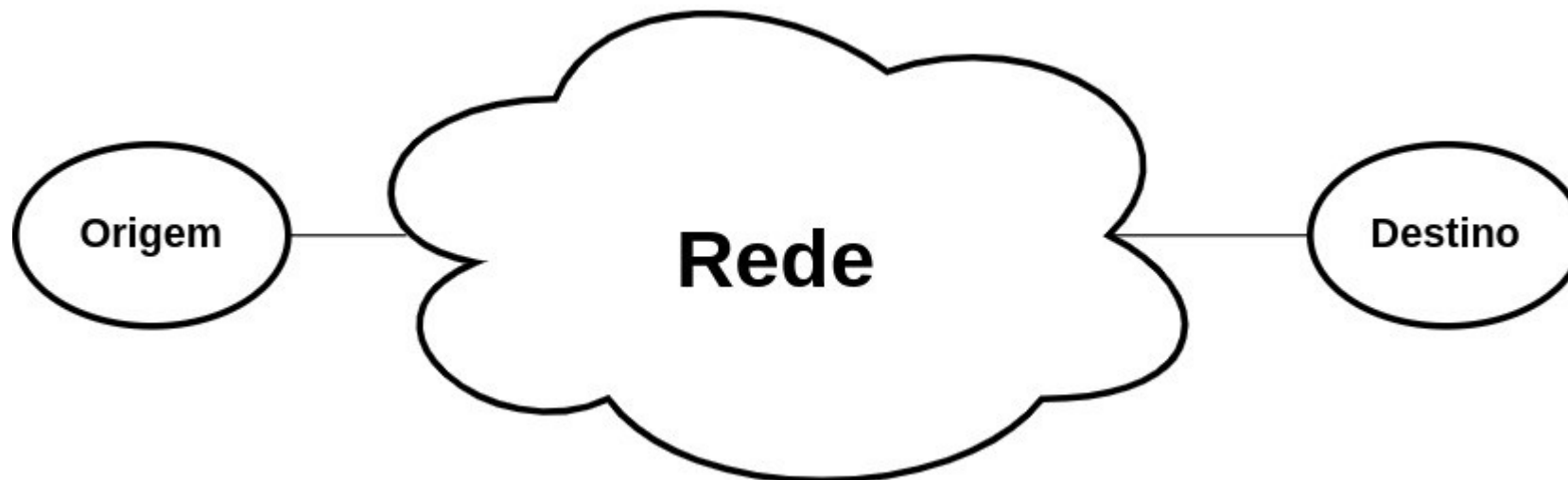
CONTROLE DE CONGESTIONAMENTO

- O controle de fluxo permite à origem estimar a capacidade do destino:



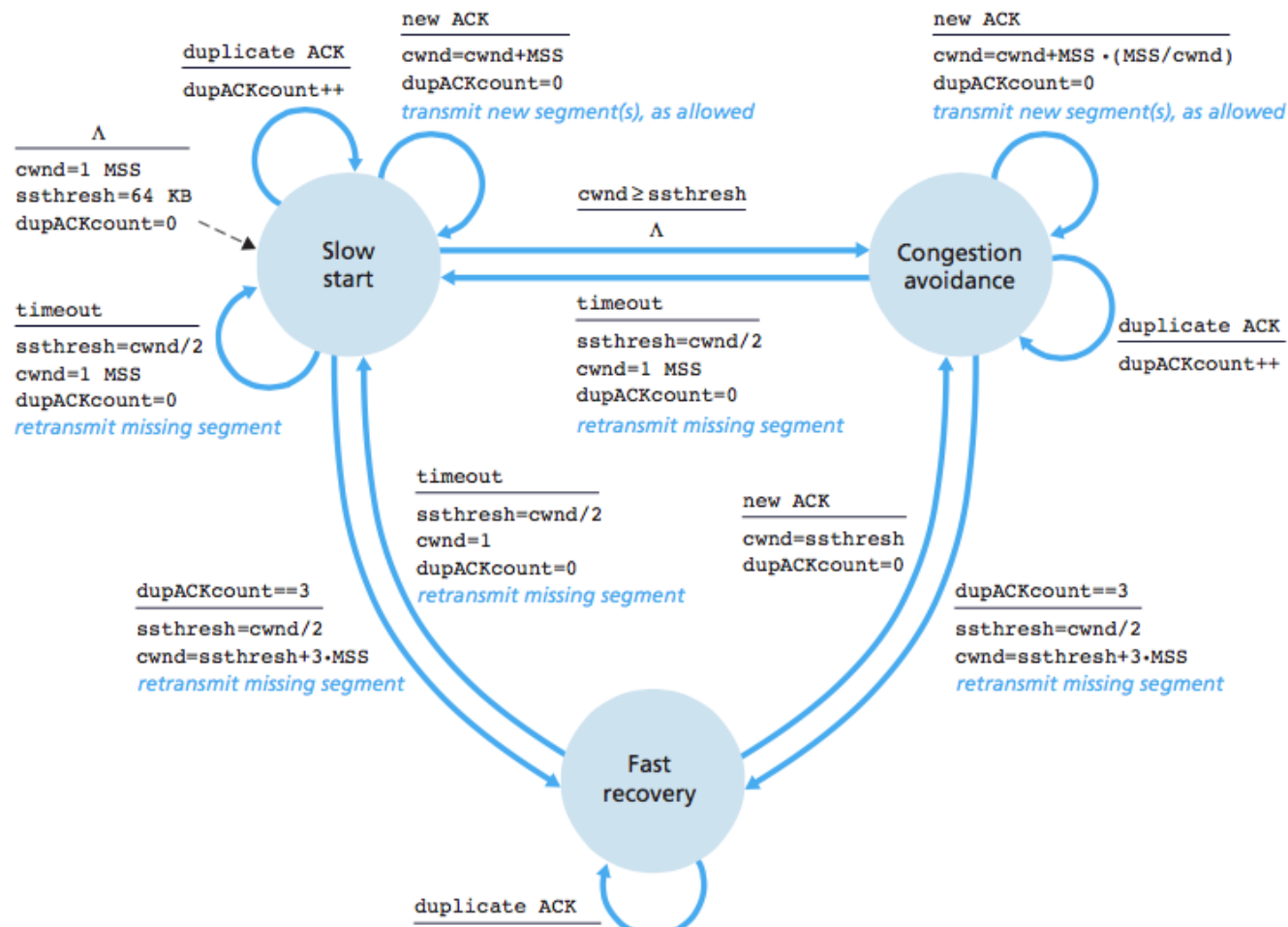
CONTROLE DE CONGESTIONAMENTO

- O controle de fluxo permite à origem estimar a capacidade do destino:
- Entre a origem e destino: a rede!
- O controle de congestionamento permite à origem estimar a capacidade da rede



| | | | |
|-----------------------------|--|---|--|
| Slow Start | Um mecanismo de controle de congestionamento que inicia a transmissão de dados em uma taxa baixa e aumenta exponencialmente até detectar perda de pacotes. | Rápido aumento da taxa de transmissão até o primeiro sinal de congestionamento, minimizando o tempo ocioso da rede. | Tahoe, Reno, New Reno, Vegas, BBR |
| Congestion Avoidance | Após o slow start, aumenta linearmente a taxa de transmissão para evitar congestionamento. | Evita o congestionamento pela moderação do crescimento da taxa de transmissão conforme a capacidade da rede é aproximada. | Tahoe, Reno, New Reno, Vegas, BBR, CUBIC |
| Fast Retransmit | Reduz o tempo necessário para recuperar um pacote perdido ao retransmitir imediatamente após detectar várias duplicatas de ACKs. | Melhora o desempenho em condições de perda de pacotes, reduzindo a latência de recuperação. | Tahoe, Reno, New Reno |
| Fast Recovery | Após uma retransmissão rápida, diminui a janela de congestionamento apenas | Permite uma recuperação mais rápida do congestionamento | Reno, New Reno, CUBIC |

| | | | |
|---|--|--|-----------------------------|
| Explicit Congestion Notification (ECN) | Permite que os equipamentos de rede notifiquem o remetente sobre o início do congestionamento antes que a perda de pacotes ocorra. | Permite ajustes proativos no controle de congestionamento, reduzindo a perda de pacotes e melhorando o throughput geral. | DCTCP, ocoptionalBBR e C |
| Selective Acknowledgment (SACK) | Permite ao receptor informar ao remetente exatamente quais pacotes foram recebidos com sucesso, permitindo a retransmissão seletiva de pacotes perdidos. | Aumenta a eficiência em ambientes com alta taxa de perda de pacotes ao evitar retransmissões desnecessárias. | Reno, N Reno, S Vegas |
| Proportional Rate Reduction (PRR) | Algoritmo que ajusta a taxa de envio de dados após uma perda de pacote para coincidir com a taxa na qual a rede pode transmitir dados com sucesso. | Reduz a latência de recuperação após perda de pacote e melhora a utilização da rede em cenários de congestionamento. | CUBIC, I |



RELACIONAMENTO DE MECANISMOS DE CONGESTIONAMENTO

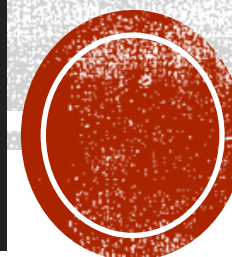
Relação entre mecanismos e trigger



| Algoritmo | Ano | Principais Características | Contexto de Uso Ideal | Vantagens | Desvantagens |
|--------------|------|---|--|---|---|
| TCP Tahoe | 1988 | Início lento, evitação de congestionamento e retransmissão rápida | Redes iniciais de baixa velocidade e confiabilidade | Implementação simples e robusta | Reage lentamente a mudanças de estado da rede |
| TCP Reno | 1990 | Recuperação rápida adicionada ao Tahoe | Redes com confiabilidade moderada e latência estável | Melhora na gestão de uma única perda de pacote | Problemas com várias perdas consecutivas |
| TCP New Reno | 1999 | Recuperação rápida aprimorada | Ambientes com perda múltipla e conexões longas | Eficiente em cenários de múltiplas perdas de pacote | Complexidade de controle aumentada |
| TCP Vegas | 1994 | Deteção proativa de congestionamento | Redes com boas condições de | Evita congestionamentos antes de ocorrerem | Desempenho varia com a precisão do |

OS
NAIS

s mais



| | | | | | |
|----------------------|------|--|---|---|---|
| TCP SACK | 1996 | Reconhecimento seletivo, otimiza a retransmissão | Redes com condições variáveis e perdas esporádicas | Eficiente em recuperar múltiplas perdas de pacote | Mais custoso em termos de cabeçalhos de pacotes |
| HighSpeed TCP | 2003 | Janela de congestionamento cresce rapidamente | Redes de longa distância e alta capacidade | Utiliza eficientemente grandes larguras de banda | Pode ser agressivo demais para redes menores |
| CUBIC | 2008 | Escala a janela de congestionamento usando uma curva cúbica | Redes de banda larga com alta capacidade | Melhor desempenho em redes de alta velocidade | Pode ser ineficaz em redes de baixa capacidade |
| TCP BBR | 2016 | Baseia-se na medição de largura de banda e RTT, independente de perdas | Redes modernas de alta velocidade e internet global | Consistência e eficiência em diferentes condições de rede | Pode enfrentar problemas com variações de rede persistentes |

M O S
O N A I S

os mais



O algoritmo de BBR contém quatro fases: *Startup*, *Drain*, *ProbeBandwidth* e *ProbeRTT* [23].

1. *Startup*:

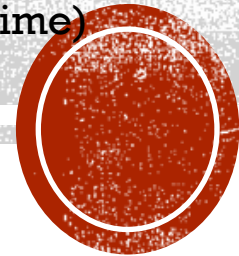
- a. *Slow start* (New Reno);
- b. Para cada pacote enviado é calculada uma amostra de RTT;
- c. $RTTP = RTTmin$ (nos últimos dez segundos);
- d. Análise da taxa de entrega para cada ACK recebido. Caso o crescimento da taxa de entrega fique estagnado por três RTTs consecutivos (indicador de descoberta do gargalo da largura de banda da rede pelo BBR), o mesmo vai para fase de *Drain*.

2. *Drain*:

- a. Para cada pacote enviado é calculado uma amostra de RTT;
- b. $RTTP = RTTmin$ (nos últimos dez segundos);
- c. Redução da taxa de entrega para uma parcela do produto da estimativa periódica da largura de banda disponível (BWBTL) e Tempo de propagação de ida e volta (RTTP);
- d. O valor *bandwidth delay product*²² (BDP) é encontrado através da equação $BDP = (RTTP \times BWBTL)$;
- e. $CWND = BDP$.

BBR

Algoritmo BBR (Bottleneck
Bandwidth and Round-trip
propagation time)



3. *ProbeBandwidth*:

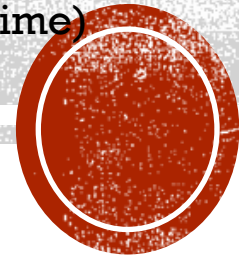
- a. Fase de maior duração no fluxo, ativada periodicamente durante todo período de transmissão;
- b. Propositalmente a taxa de entrega é alterada para um valor maior que o valor BDP;
- c. Volta para a fase *Drain*, com valores reduzidos de taxa de entrega para uma parcela do BDP.

4. *ProbeRTT*:

- a. Fase que é ativada caso o valor do RTTP fique inalterado por dez segundos durante a transmissão;
- b. Interrupção do *ProbeBandwidth*.
- c. Ocorre uma remediação do RTTP.

BBR

Algoritmo BBR (Bottleneck Bandwidth and Round-trip propagation time)



- I. Se ocorrer recebimento de três ACKs duplicados:
 - a. Não uso do algoritmo de *Slow Start*;
 - b. $ssthresh = (CWND / 2)$;
 - c. Retransmissão do segmento ausente com o *Fast Retransmit*;
- II. *Artificial Inflation*¹⁸ com *Fast Recovery*:
 - a. $CWND = ssthresh + 3 * MSS$;
 - b. A cada ACK duplicado do segmento que ocasionou o *Fast Retransmit*,
 $CWND = CWND + 1$;
- III. Após às ocorrências de novos ACKs recebidos com sucesso com uso do *Fast Recovery*:
 - a. $CWND = ssthresh$ (com o valor do momento da perda);
 - b. Encerramento do uso do algoritmo *Fast Recovery*;
 - c. Execução do *Congestion Avoidance* com AIMD, com aumento da $CWND$ maneira linear, mas diminuindo exponencialmente.

RENO

Algoritmo RENO



Onde,

- C é uma variável escalar auxiliar da função Cubic;
- T é tempo da redução da janela de congestionamento;
- $CWND_{máx}$ é tamanho da janela antes da redução;
- $CWND$ é o valor que definirá como ocorrerá o crescimento da janela;
- K é período necessário para função conseguir chegar em $CWND_{máx}$ sem a ocorrências de eventos de perdas. O cálculo da variável K é mostrado Equação 2:

$$K = \frac{\sqrt[3]{CWND_{máx} \cdot \beta}}{C}$$

Equação 2 – Período necessário para atingir $CWND_{máx}$.

- β é fator de redução multiplicativo.

CUBIC

Função Cúbica



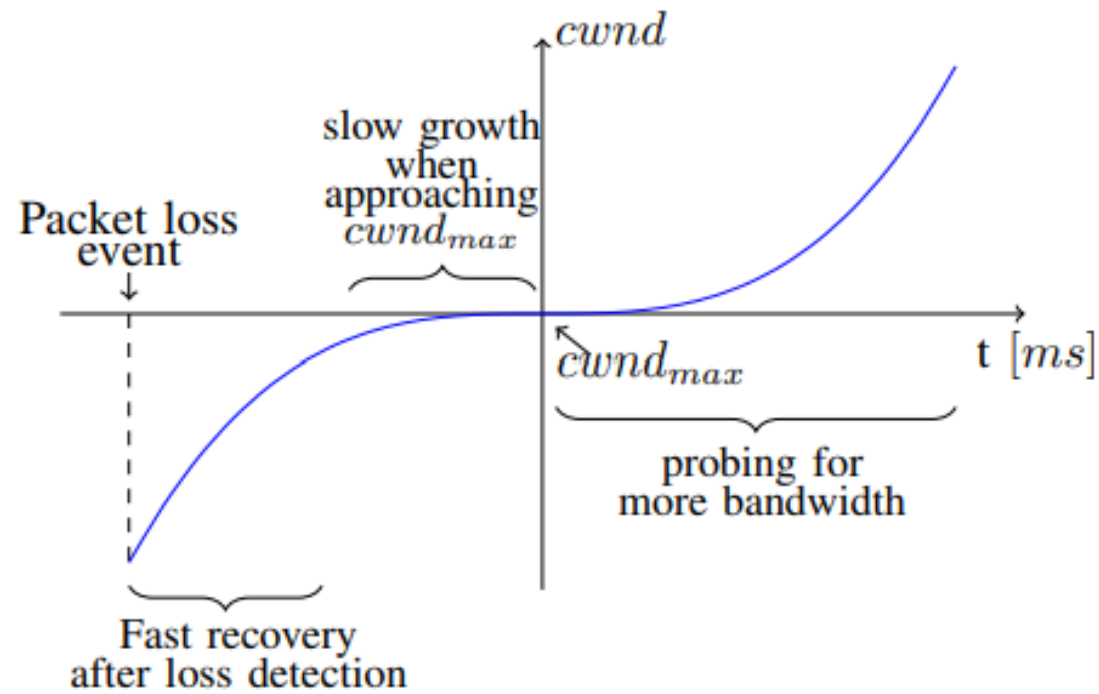
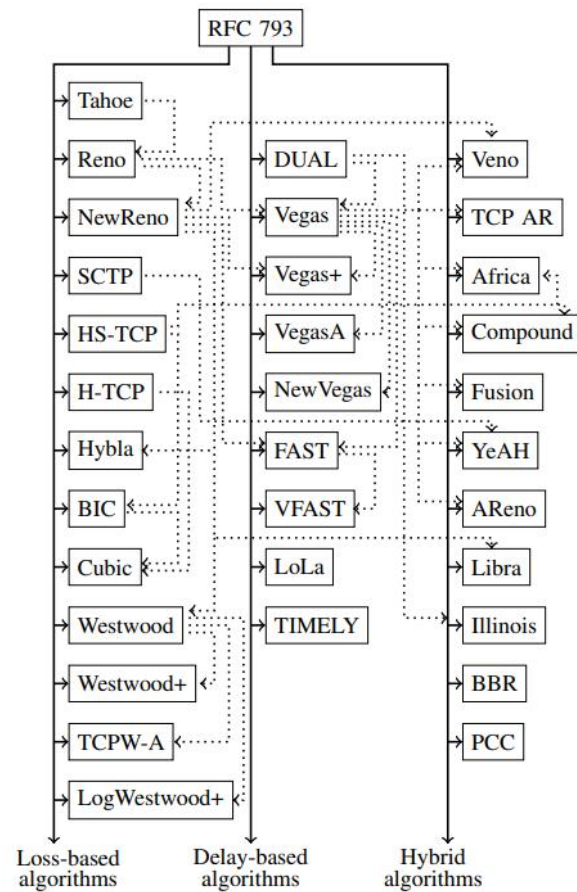


Fig. 4: Cubic function used for the congested window size ($cwnd$).

CUBIC

Função Cúbica

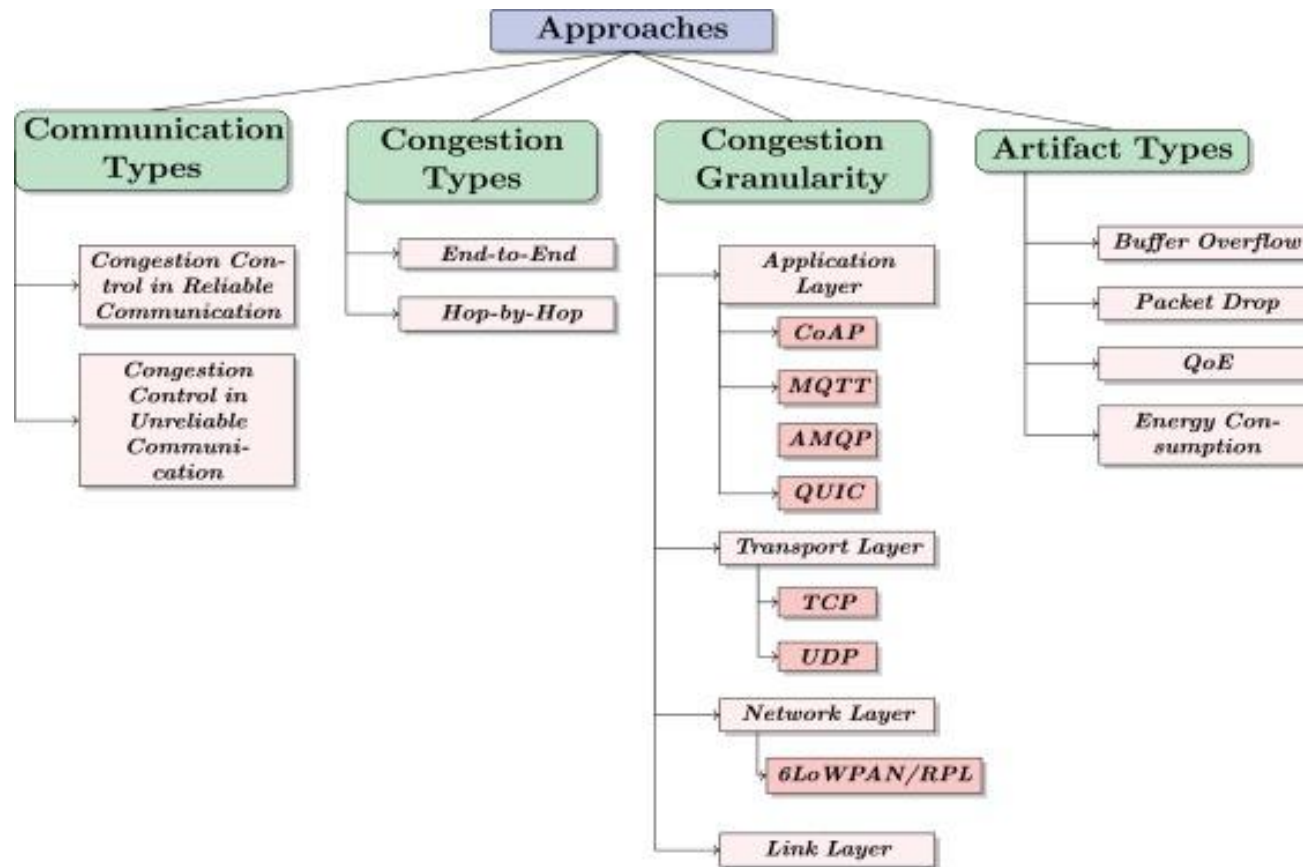




TAXONOMIA

Taxonomia por Loss,
Hybrid and Delay

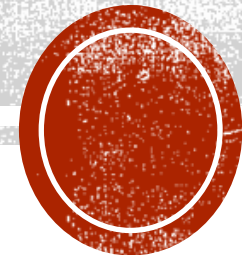


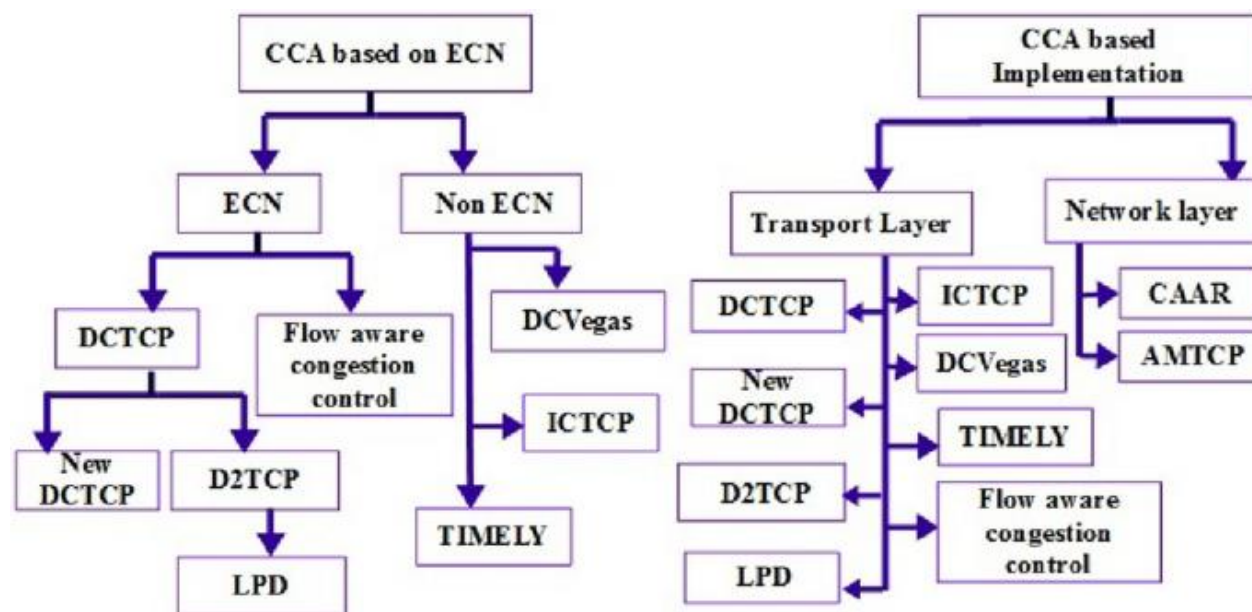


1

CONGESTIONAMENTO
EM RELAÇÃO AO
CONTEXTO

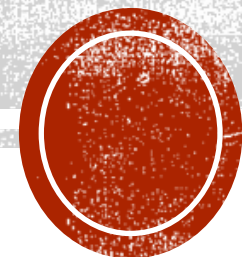
Congestionamento e IOT

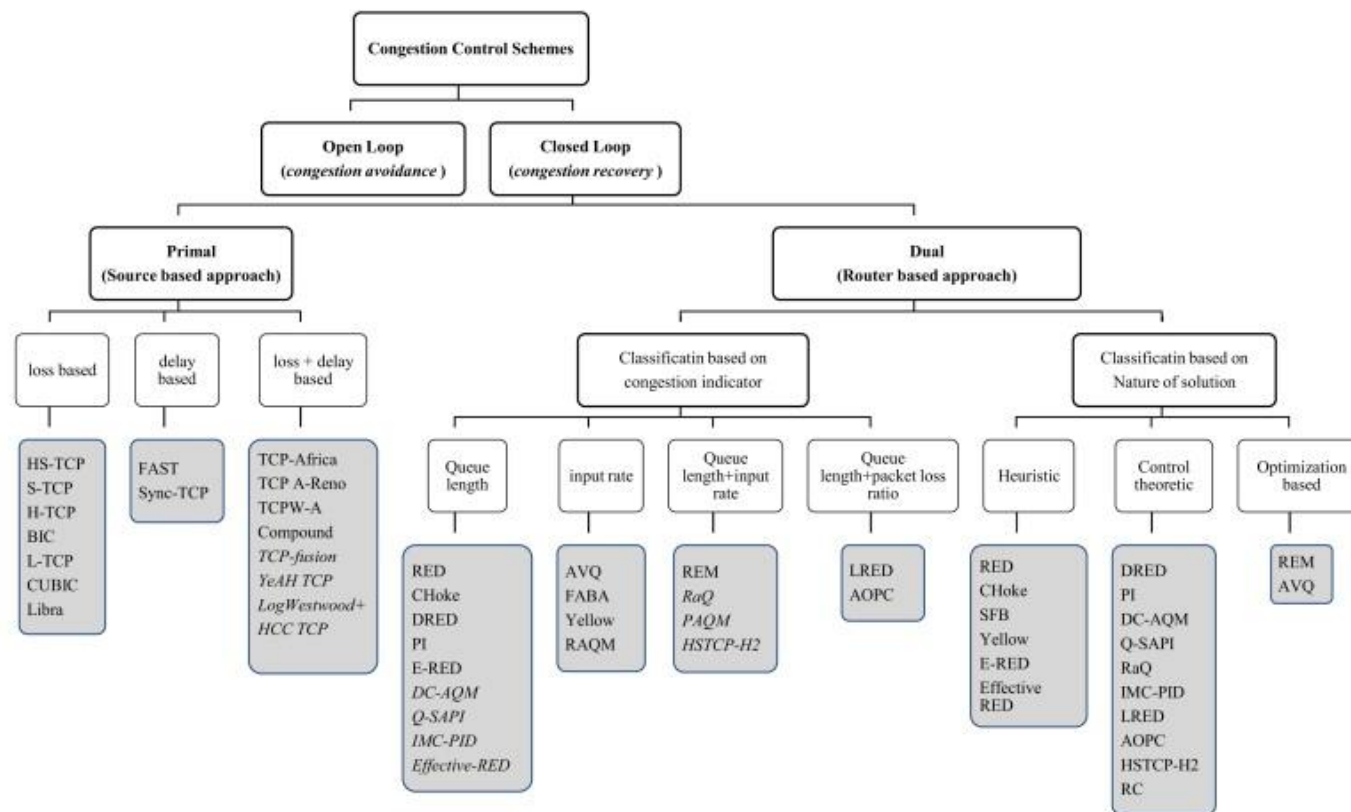




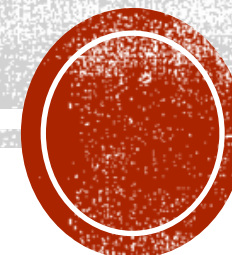
CONGESTIONAMENTO
EM RELAÇÃO AO
CONTEXTO

Congestionamento e Cloud





VISÃO GERAL





OBRIGADO!

