Control de congestión Palermo:

Prevención de bufferbloat y distribución justa de capacidad

Alejandro Popovsky
Universidad de Palermo
Facultad de Ingeniería

lacnic39
8-12 Mayo / Mérida, México

Facultad de Ingeniería

Juntos transformemos Yucatán
GOBIERNO DEL ESTADO

Algoritmos comunes de ctrl de congestión

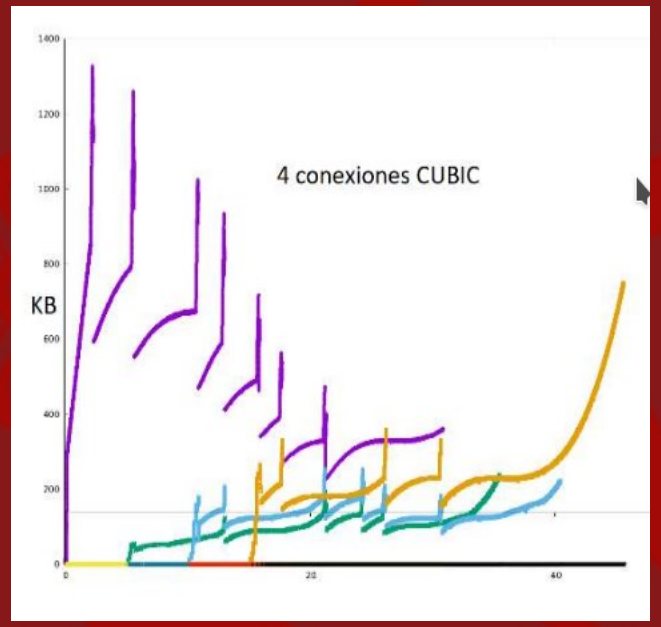
Objetivos:

- Maximize throughput
- Minimize losses

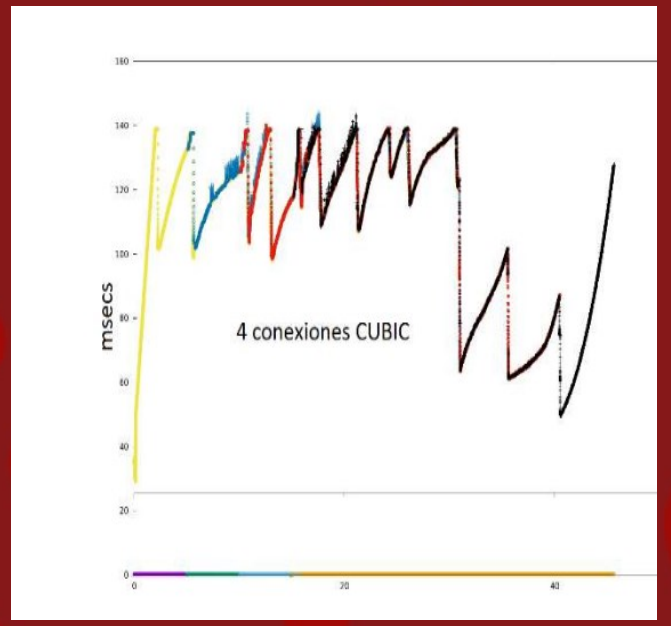
Basados en detección de pérdidas

Reno / Cubic

Problemas de los algoritmos comunes de ctrl de congestión



Evolución de
Datos in-flight



Evolución de
Round trip time

Efectos del buffer bloat

Aumento de latencia para conexiones que comparten el cuellos de botella con largas transmisiones.

Muy significativo efecto en conexiones orientadas a conexión: páginas web, requests-responses, etc.

Afecta tanto las conexiones de otros usuarios como otras conexiones del mismo usuario.

Cuello de botella compartido

Porción obtenida de la capacidad de cada conexión individual = Porción de ocupación del buffer de encolamiento de cada conexión individual

=> Pocos incentivos para la prevención del buffer bloat

Algoritmos tradicionales sensibles a la latencia

Ejemplos: LEDBAT, VEGAS, VENO, TCP-LP, …

Performance obtenida al compartir cuellos de botella:
“Less than Best Effort Congestion Control” (LBE)

Alternativas a LBE: comportamiento adaptativo

Palermo: Realimentación sobre el cuellos de botella

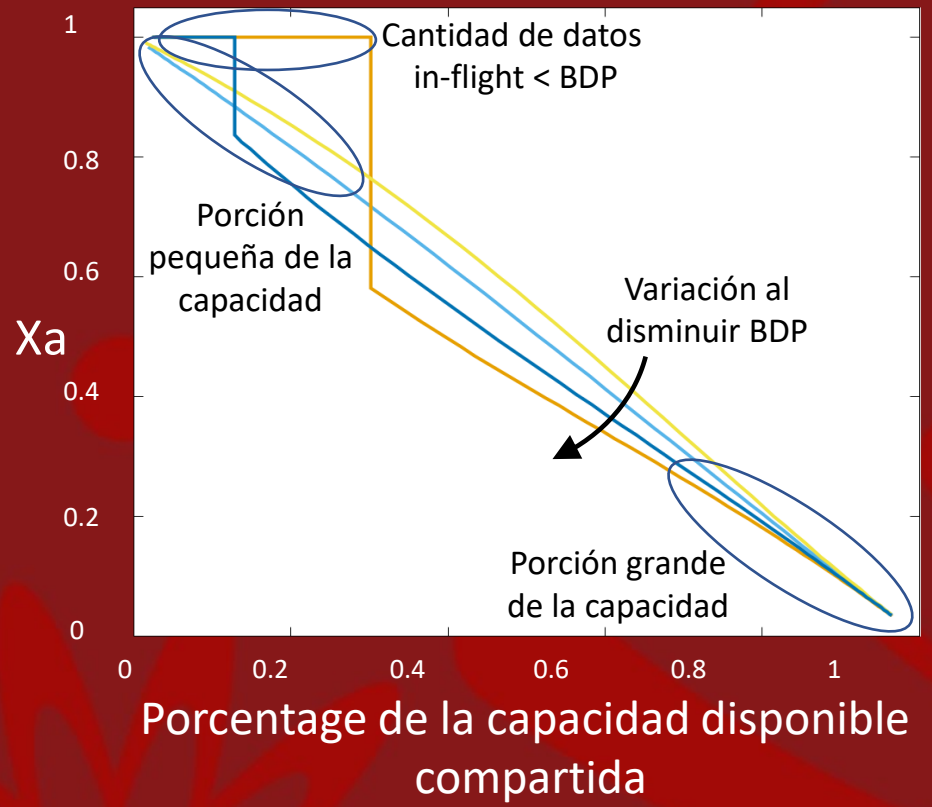
Objetivo: estimar la ***porción utilizada de la capacidad disponible***

Estimador propuesto:

Variación de ritmo proporcional Proportional (Ra) a variaciones de cantidad de datos In-flight (Ca)

$$Xa = \left(\frac{\Delta Ra}{\Delta Ca}\right)\left(\frac{Ca}{Ra}\right)$$

Palermo: Realimentación sobre el cuellos de botella



Uso exclusivo del cuello de botella:

- Cantidad In-Flight < BDP $\Longrightarrow$ Xa=1
- Cantidad In-Flight > BDP $\Longrightarrow$ Xa=0

Uso compartido del cuello de botella:

- Xa ≈ (1 – share of capacity)

BDP: product ancho de banda por delay

Controls de Congestion Palermo

Mantiene objetivos comunes:

- Optimizar throughput
- Minimizar pérdidas

Objetivos agregados:

- Minimizar latencia
- Reparto justo de capacidad disponible

Comportamiento Adaptativo:

- Prevención de buffer bloat mientras sea possible.
- Sino es possible, volver al control tradicional

Control de congestion Palermo
version para **recepción**

Desarrollo inicial en 2016, presentado IETF 95.

Oportunidad observada en los IXP’s de CABASE:

Tráfico limitado por control de flujo sin Ventana de recepción dinámica (DRS), a veces obtiene el mismo throughput que tráfico limitado por control de congesitón, pero con menores latencias.

Testeo extensivo en los proxies de la Universidad de Palermo.

Detalles del algoritmo

Cuando la capacidad disponible no está todavía alcanzada:

$\Longrightarrow$ crecimiento regular de la Ventana de congestion (CWND)

Cuando la capacidad disponible está alcanzada:

Pero con una porción pequeña $\Longrightarrow$ crecimiento regular de CWND

Pero con una porción grande $\Longrightarrow$ oscilar CWND

Criterio de adaptación:

El algoritmo obtiene una latencia primedio que aumenta con la cantidad de conexiones (desconocida) que comparten el cuello de botella. Entonces cuando esta cantidad excede un cierto límite ya beneficios y retorna al control de congestion regular (reno-cubic).

Arquitectura del control de congestión Palermo desde el emisor

Desarrollado para el Linux Kernels 5.9, y posteriores.

Módulo independiente **dinámico** para el kernel (LKM, DMKS)

Testado para arquitecturas X86_64 y extensible a otras arquitecuras como ARM, PPC, etc.

Objetivos de testeo

Chequeo de maximización throughput en condición de exclusividad en el cuello de botella

Chequeo de prevención de buffer bloat y de distribucióno justa, en condiciones de cuello de bottela compartido con conexiones bien comportadas.

Chequeo de performance en condiciones de cuello de botella compartido con conexiones mal comportadas (ctrl de congestion tradicional).

Comparación con algoritmos del estado del arte (BBR)

Chequeo de despliegue en datacenters grandes.

Control desde **receptor** versus control desde **emisor**

Desde el Receptor:

Optimiza tráfico entrante.

Imprecisión en la estimación de la limitación

Aplicación, Ctrl de flujo, ventana del sender.

Impresición en estimación del Round trip time

Necesidad de frenar al sender para retomar control

Desde el emisor:

Optimiza tráfico saliente

Mejor estimación de la limitación

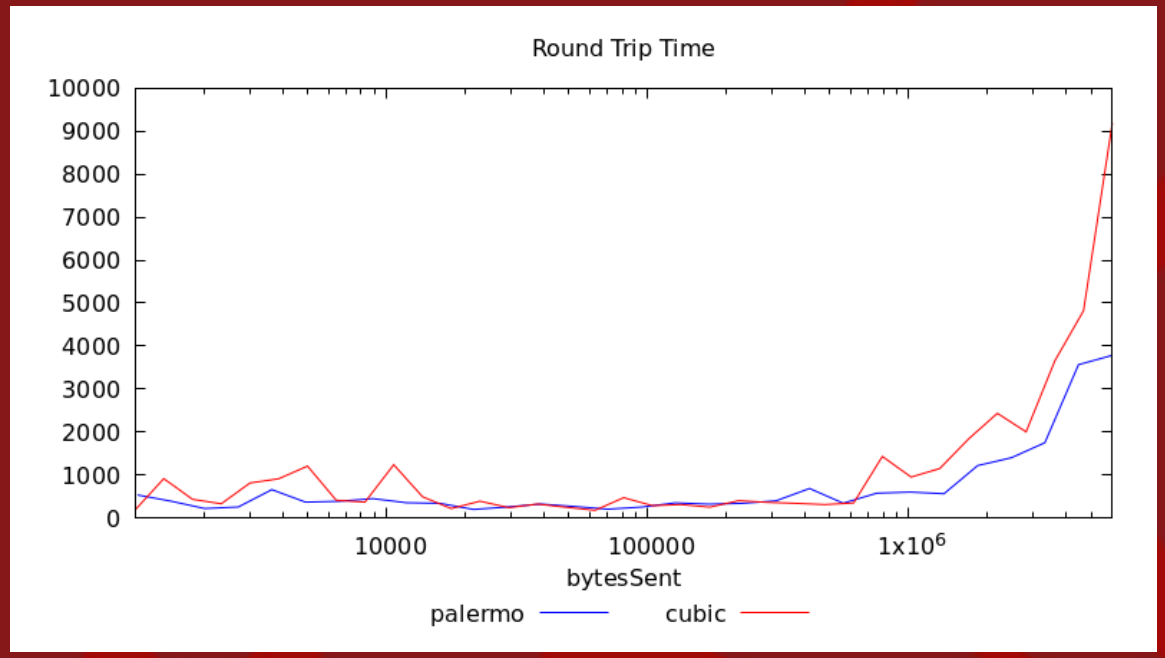
Mejor estimación de round trip time

Común a ambas estrategias de control:

Necesidad de esperar el efecto de los ajustes,

en el estimador ***Xα***

Web servers traffic



Web servers traffic: CUBIC vs PALERMO

Transmisiones cortas:

throughput y latencias similares

Transmisiones largas:

20% menos throughput a cambio de 50% menos round trip time

Gran beneficio para transacciones cortas que comparten proveedor de red con transferencias largas.

(ejemplo transacciones cortas: páginas web compuestas por muchos gráficos y objetos)

Instalación de Control de congestión Palermo

Download, compilación e instalación:

```
wget https://www.palermo.edu/ingenieria/ingenieria-telecomunicaciones/tcp/tcp_palermo-2.06.tgz
tar xvzf tcp_palermo-2.07b.tgz
cd tcp_palermo-2.07b/
make modules
make modules_install
```

Activar específicamente para un socket:

```
setsockopt(fd,SOL_TCP,TCP_CONGESTION,”palermo”,7);
```

Activar para todo el Sistema (todos los nuevos sockets):

```
modprobe tcp_palermo
echo palermo > /proc/sys/net/ipv4/tcp_congestion_control
```

Conclusiones y trabajo futuro

Control de Congestión Palermo:

- Opción válida para mejorar la performance de trafico saliente de servidores.

Próximos pasos:

- Exploración de robustez y variantes
- Pruebas a gran escala en datacenters
- Propuesta para distribución estándar del kernel de linux

Para más información o fuentes:

https://www.palermo.edu/ingenieria/ingenieria-telecomunicaciones/control-de-congestion-palermo.html

apopov@palermo.edu

Agradecimientos

Este trabajo fue financiado por LACNIC (Registry for Internet Addresses for Latin America and the Caribbean), y ganador del premio FRIDA 2020 otorgado otorgado por el Fondo Regional para Innovación en América Latina y el Caribe

lacnic39
8-12 Mayo / Mérida, México

UP Universidad de Palermo
Facultad de Ingeniería

lacnic

Juntos transformemos Yucatán
GOBIERNO DEL ESTADO

Acknowedgments

This work was supported by LACNIC (Registry for Internet Addresses for Latin America and the Caribbean), and winner of the FRIDA 2020 prize awarded by the Regional Fund for Digital Innovation in Latin America and the Caribbean.