



- Bandwidth Management -

BURST

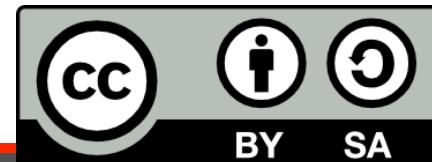
by: Novan Chris

Citraweb Nusa Infomedia, Indonesia

www.mikrotik.co.id



INDONESIA INTERNET
EXPO & SUMMIT
22 - 24 NOVEMBER 2016
BALAI KARTINI, JAKARTA



BY SA

Pokok Permasalahan

- Jaman semakin modern, banyak aspek kehidupan yang segalanya serba online :
 - Beli Barang – Online
 - Mau naik ojeg – Online
 - Beli makan – Online
 - Mau Pijat – Online
 - Mau Curhat - Online
 - Cari Jodoh ... Online
- Jika tidak online dianggap tidak modern ... katrok !

Pola Hidup Manusia Modern



- 4 Sehat 😊 -



- 5 SEMPURNA -

BANDWIDTH

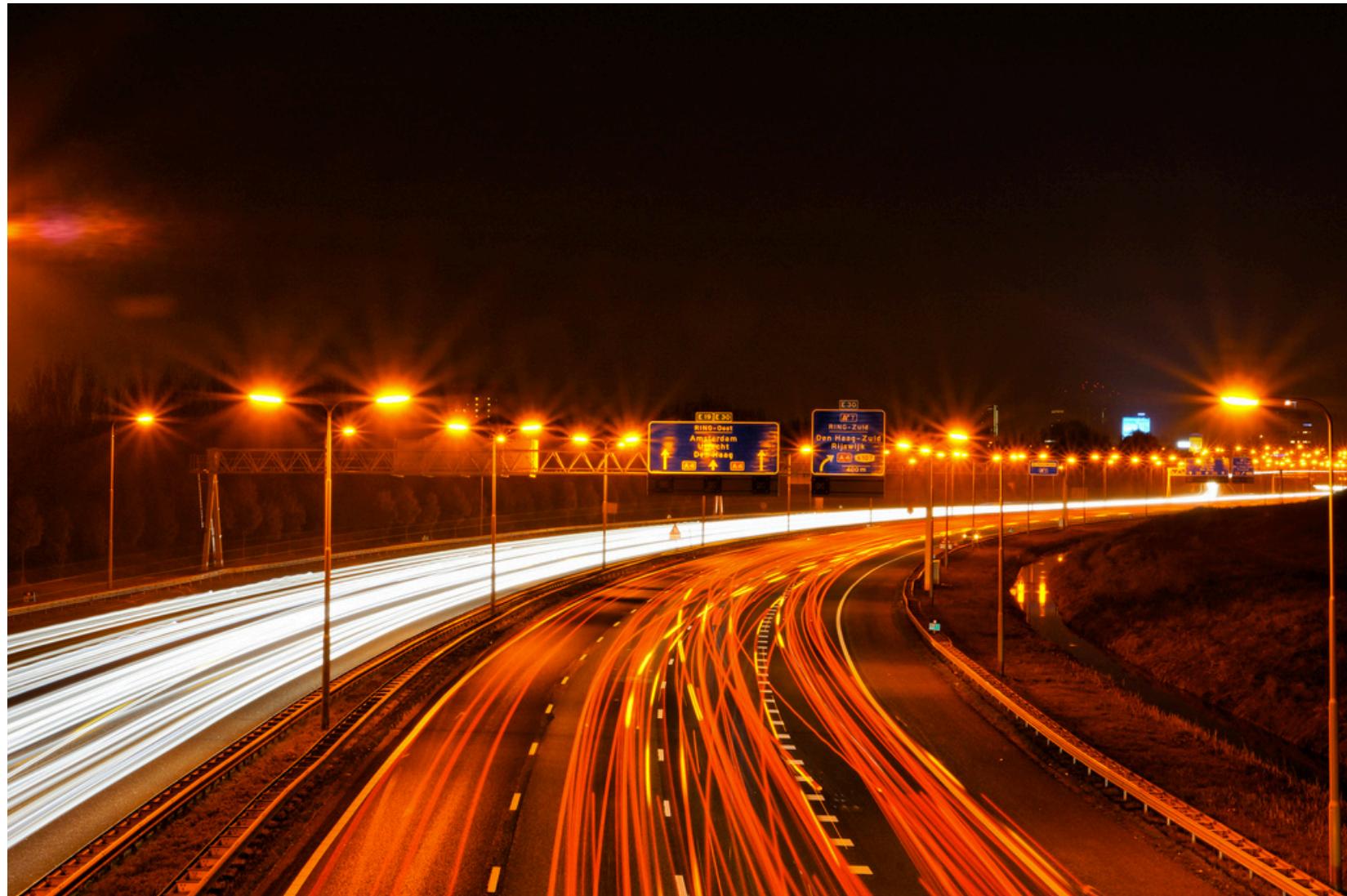


Sebesar-besarnya bandwidth tidak
akan pernah cukup memuaskan
Anda !

Kebutuhan Bandwidth

- Semakin banyak content yang ada, maka berimbang pada kebutuhan bandwidth
- Setiap user berusaha mencari bandwidth tercepat dan juga termurah ☺
- Provider dituntut untuk memberikan pelayanan sebaik mungkin kepada user
- Fungsi Bandwidth Management yang ada di Mikrotik dituntut untuk bisa memberikan kecepatan dan juga kenyamanan

Burst

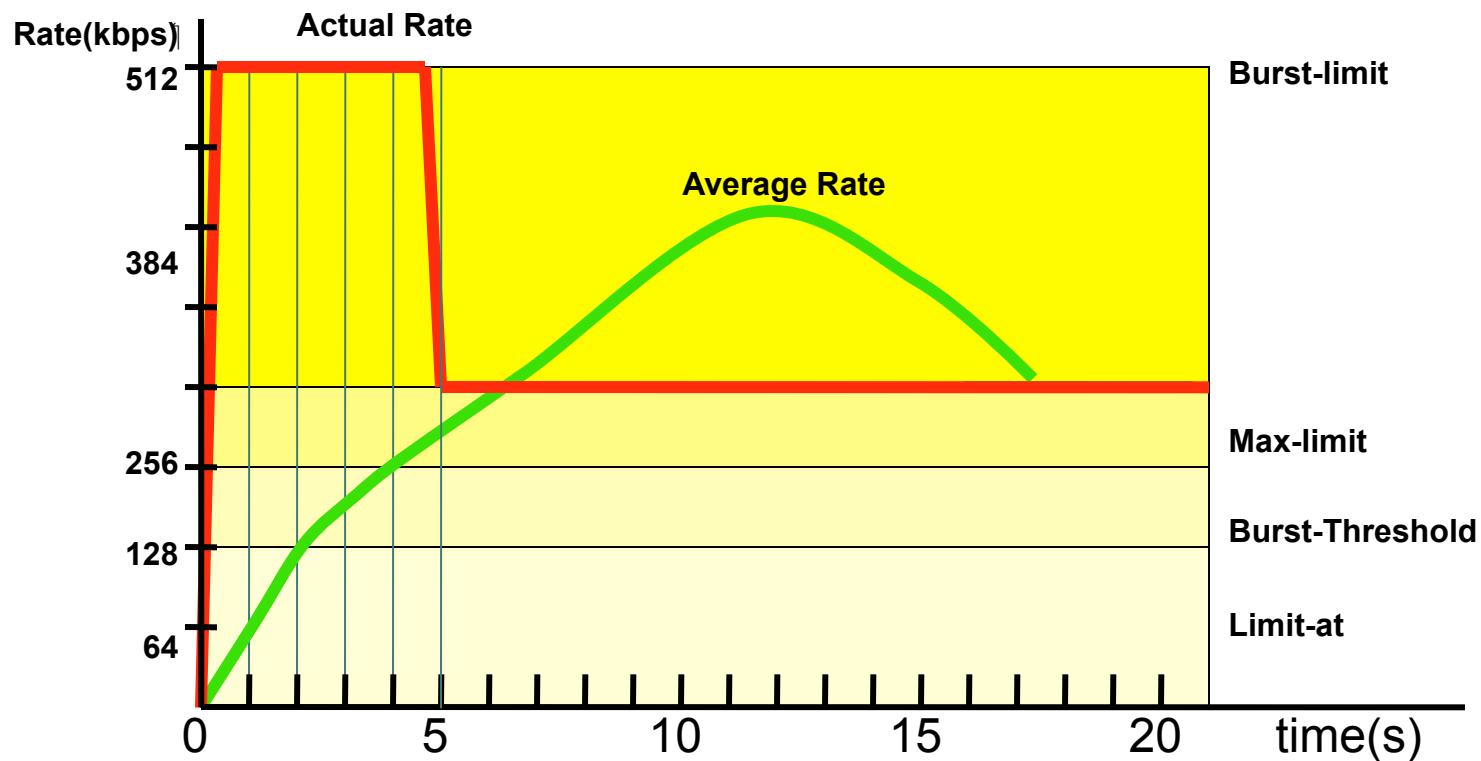


BURST

- Salah satu metode implementasi dari Bandwidth Shaper.
- Memungkinkan memberikan alokasi bandwidth yang lebih besar (dengan syarat) :
 - Ketika ada bandwidth sisa
 - Ketika kondisi sesuai dengan kriteria
- Kriteria BURST dikalkulasi dari 3 parameter :
 - Burst Limit
 - Burst Threshold
 - Burst Time

Contoh BURST

**Max-limit=256kbps, burst-time=16,
burst-threshold=128kbps, burst-limit=512kbps.**



Kalkulasi BURST

- Pada awalnya, data rate rata-rata dalam 16 detik terakhir adalah 0 kbps. Karena data rate rata-rata ini lebih kecil dari burst-threshold, maka burst dapat dilakukan.
- Setelah 1 detik, data rate rata-rata adalah $(0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+512)/16=32\text{ kbps}$, masih lebih kecil dari **burst-threshold**. Burst dapat dilakukan.
- Untuk detik kedua, data rate rata-rata adalah $(0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+512+512)/16=64\text{ kbps}$.
- Demikian pula untuk detik ketiga, data rate rata-rata adalah $(0+0+0+0+0+0+0+0+0+0+0+0+0+0+0+512+512+512)/16=96\text{ kbps}$.
- Setelah 4 detik, tibalah pada saat di mana data rate rata-rata lebih besar dari **burst-threshold**. Burst tidak dapat lagi dilakukan, dan data rate turun menjadi **max-limit** (256kbps).

Rumus BURST

$$\text{Burst Limit} : \text{Burst Threshold} = \text{Burst Time} : \text{Client Burst}$$

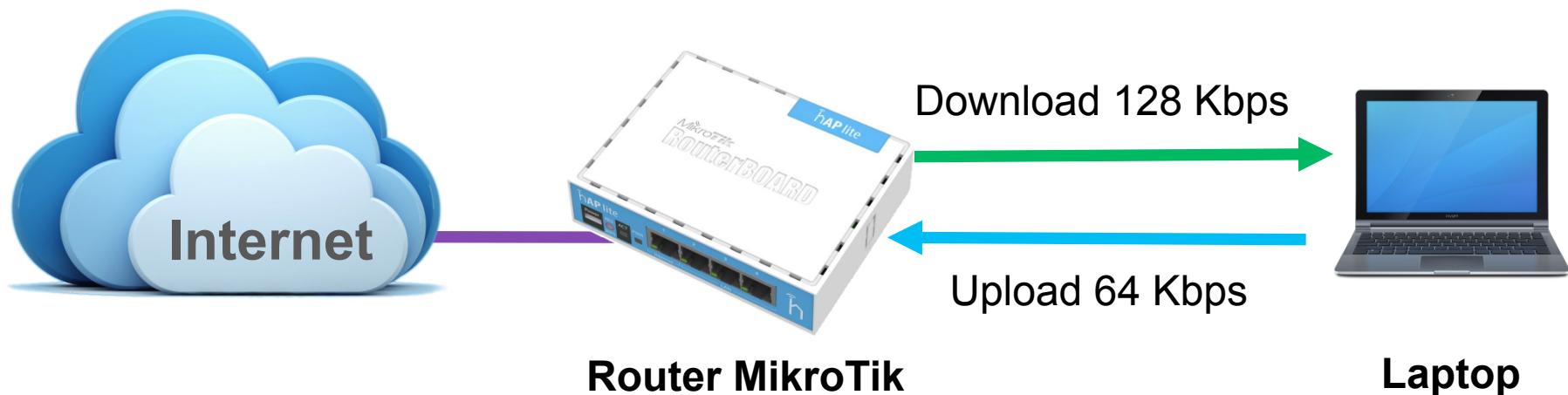
**Max-limit=256kbps, burst-time=16,
burst-threshold=128kbps, burst-limit=512kbps.**

$$128 / 512 * 16 = 4 \text{ detik}$$

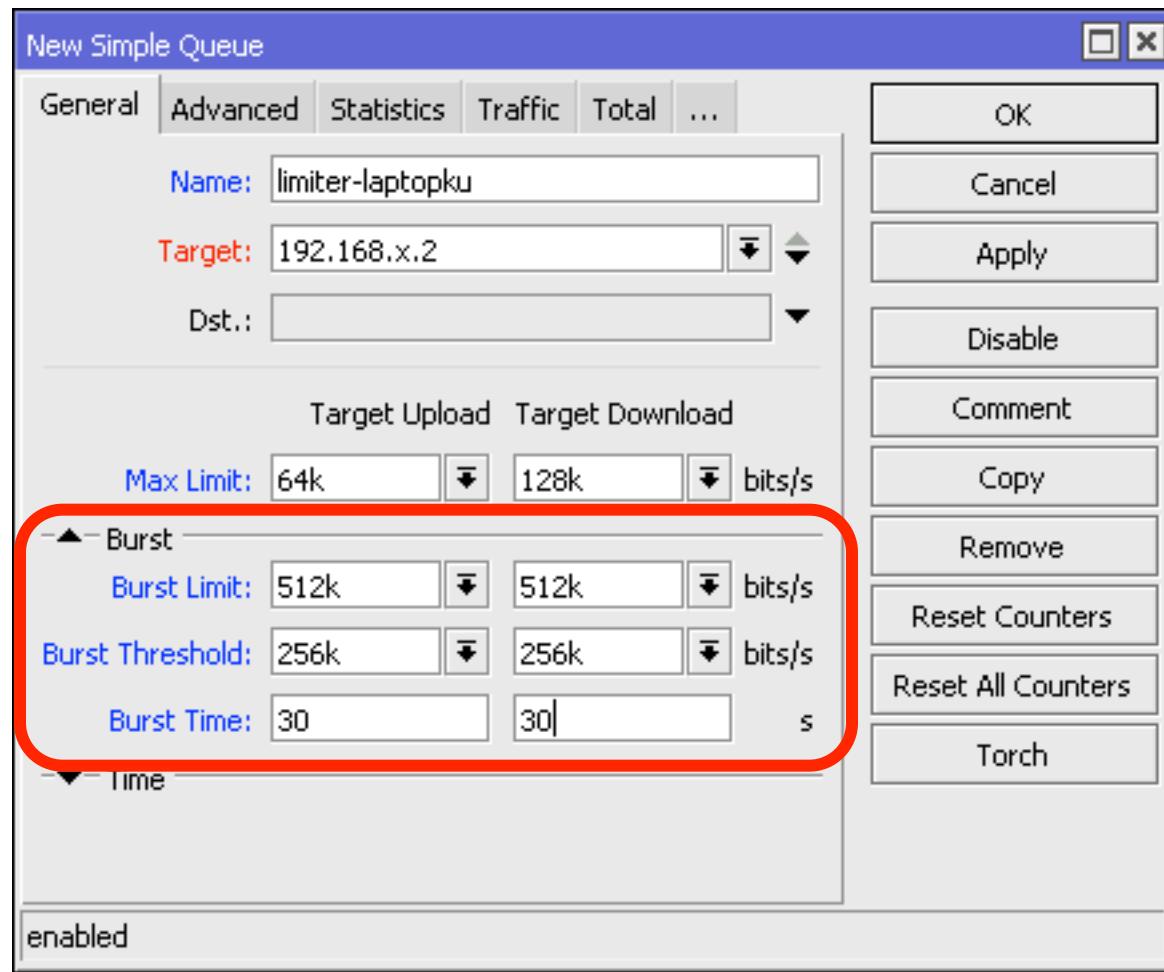
- Jika burst-threshold dibawah max-limit maka client hanya bisa burst satu kali, jika burst-threshold di atas max-limit client bisa burst berkali-kali.
- Rumus di atas hanya berlaku pada kondisi awal ketika rata-rata penggunaan 0 bps

Contoh Burst

Tambahkan ***burst-limit=512K***,
burst-threshold 256k dan
burst-time=30s



Burst Config



Proses Burst yang berat

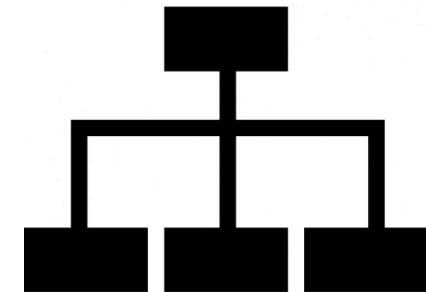
- Proses Burst melakukan kalkulasi berdarakan 16 slot waktu :
 - Burst-time=16s – slot-waktu=1s
 - Burst-time=32s – slot-waktu=2s
 - Burst-time=10s – slot-waktu=0.625s
- Router harus mengingat penggunaan bandwidth real dari client selama 16 slot waktu
- Router harus mengkalkulasi rata-rata penggunaan bandwidth real per 16 slot waktu

Burst Alternative

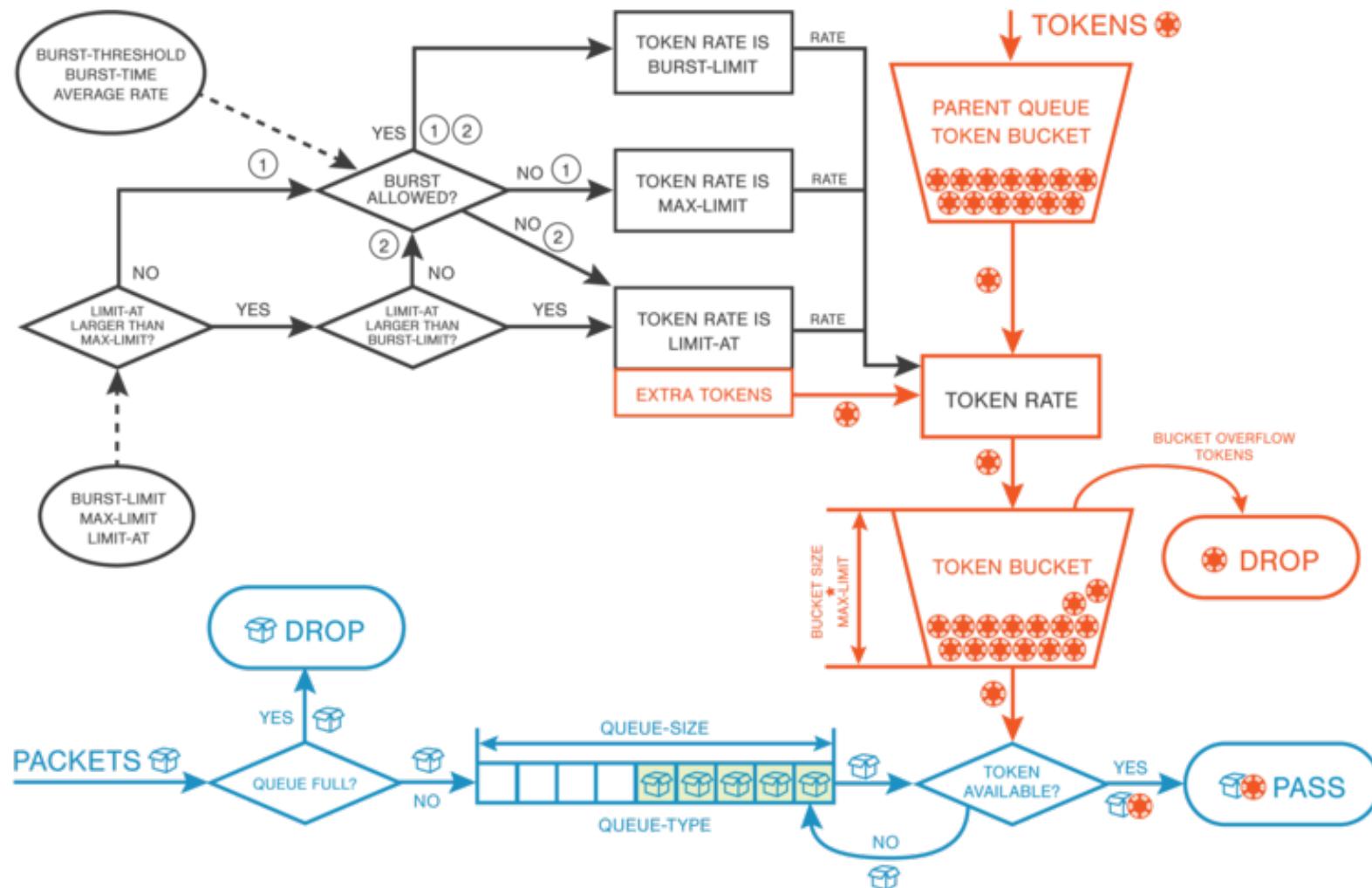
- Mekanisme fitur Burst semakin membebani processor ketika bandwidth yang dimanage bertambah besar.
- Mikrotik akhirnya membuka metode baru yang sebelumnya tersembunyi yaitu metode Token Bucket sebagai metode alternatif dari Burst yang lebih sederhana.

HTB – Hierarchical Token Bucket

- Cara kerja Hirarki pada HTB sudah banyak dibahas.
 - [http://mikrotik.co.id/
artikel_lihat.php?id=29](http://mikrotik.co.id/artikel_lihat.php?id=29)
- Mengenai Token Bucket ini sendiri mungkin masih belum banyak dipublikasikan
- Fungsi Token Bucket sudah ada sejak dulu menjadi satu pada HTB.



Token Bucket Algorithm



Token Bucket Algorithm

- Token Bucket memiliki algoritma tersendiri untuk mengolah bandwidth di tiap satu rule queue.
 - sub-process pada HTB
- Algoritma Token Bucket menggunakan analogi ticket/token yang bisa ditampung pada sebuah buffer (Bucket)
 - **Token** dihitung dalam satuan bytes per second
 - Sedangkan **Bucket** adalah buffer yang dikalkulasi dari max-limit

Token Bucket Logic



- Token Rate - Extra Bandwidth
- Token Rate - Extra Bandwidth
- Token Rate - Extra Bandwidth



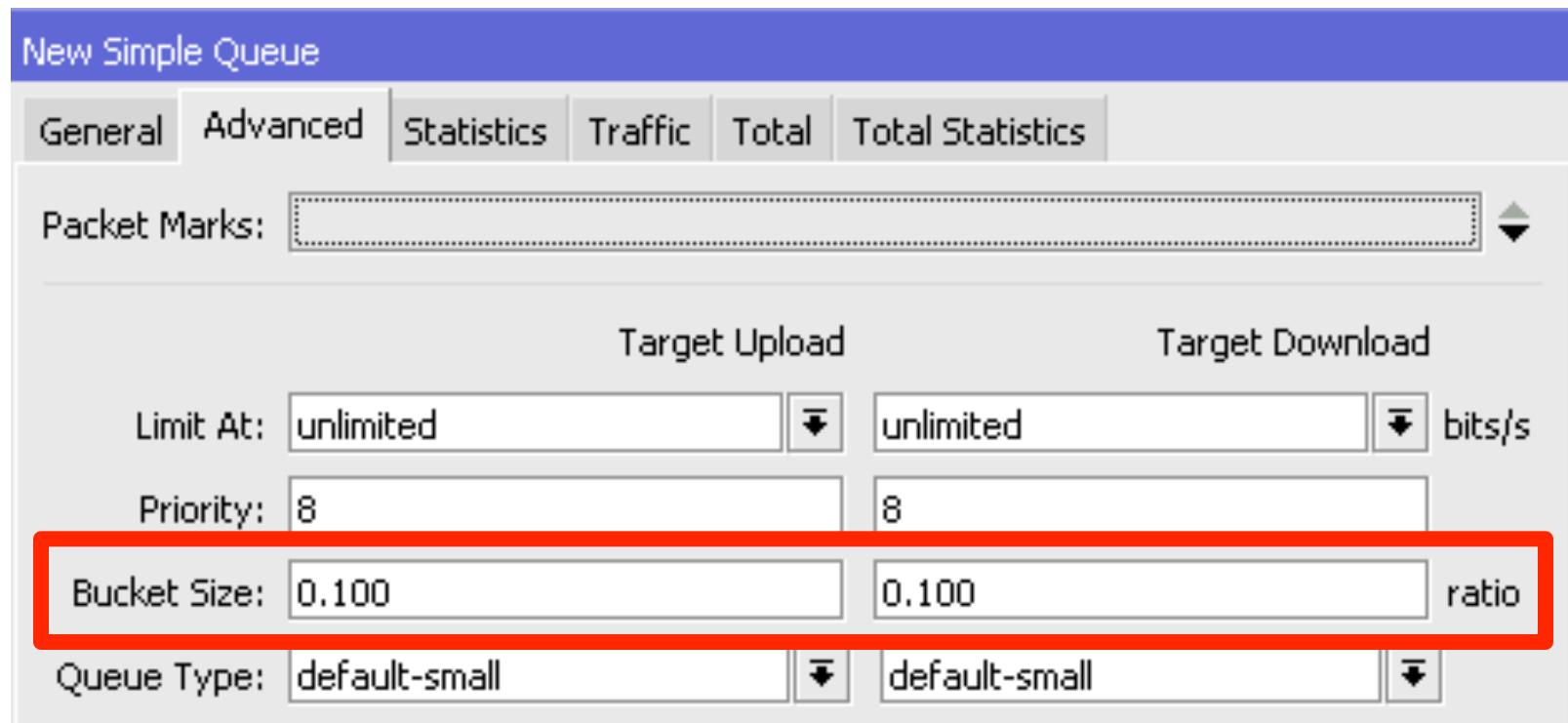
- Empty Bucket -
No token rate
No extra bandwidth



Token Bucket Parameter

- **Token Rate** – dikalkulasi dari limitasi terbesar dalam satu rule queue :
 - **Limit-at** (jika limit-at lebih besar dari max-limit parent)
 - **Max-limit**
 - **Burst-limit** (jika burst-limit aktif dan kondisinya diperbolehkan untuk burst)
- **Bucket Capacity = bucket-size * max-limit**
 - Dari dulu parameter bucket-size dikunci pada nilai 0.1
- Mulai versi **6.35** kita bisa mengatur parameter ini

bucket-size configuration



Contoh Kalkulasi – default setting

- Terdapat Queue :
 - Queue simple add max-limit=20M/20M target=172.16.1.212/32
- Kalkulasi Bucket-capacity :
 - bucket-size=0.1
 - Max-limit=20M
 - $0.1 * 20M = 2M$
- Dengan bucket-capacity 2Mbit maka client bisa menyalurkan data 2Mbps tanpa ada limitasi
- Karena bucket-capacity masih dibawah max-limit maka hal ini tidak ada efeknya

Contoh Kalkulasi

- Modifikasi rule menjadi :
 - Queue simple add max-limit=20M/20M target=172.16.1.212/32
bucket-size=10
- Kalkulasi bucket-capacity :
 - bucket-size=10 & Max-limit=20M
 - $10 * 20 = \text{200M}$
- Dengan bucket-capacity 200Mbit maka client bisa menyalurkan data 200Mbps tanpa ada limitasi
 - 40Mbps selama 10s (extra 20Mbps selama 10s)
 - 100Mbps selama 2.5s (extra 80Mbps selama 2.5s)
 - 1Gbps selama 200ms

Contoh Kalkulasi

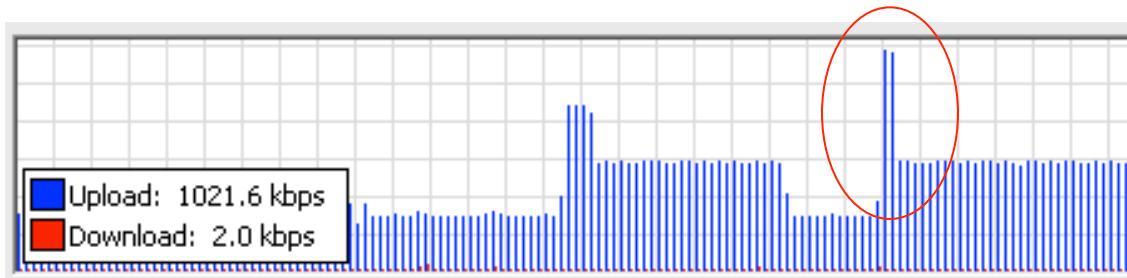
Max-limit = 20Mbps + Bucket-capacity=200M

Traffic yang digunakan 40mbps

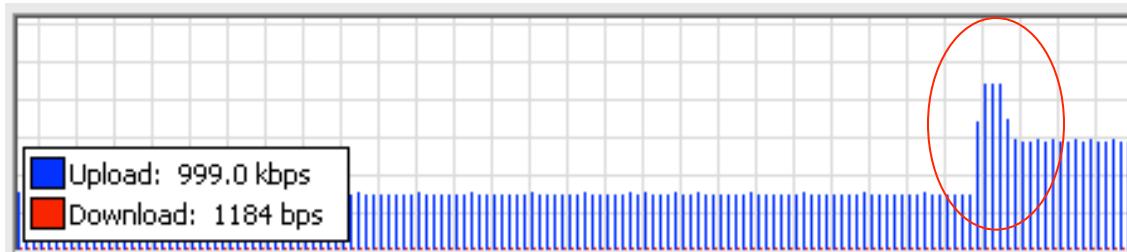
- #1 20Mbps + 20Mbps - Bucket-Capacity=180M
- #2 20Mbps + 20Mbps - Bucket-Capacity=160M
- #3 20Mbps + 20Mbps - Bucket-Capacity=140M
- #4 20Mbps + 20Mbps - Bucket-Capacity=120M
- #5 20Mbps + 20Mbps - Bucket-Capacity=100M
- #6 20Mbps + 20Mbps - Bucket-Capacity=80M
- #7 20Mbps + 20Mbps - Bucket-Capacity=60M
- #8 20Mbps + 20Mbps - Bucket-Capacity=40M
- #9 20Mbps + 20Mbps - Bucket-Capacity=20M
- #10 20Mbps + 20Mbps - Bucket-Capacity=0
- #11 20Mbps

Contoh 2

- Max-limit=1M bucket-size=2
- Bucket-capacity=2M
- Test penggunaan :
 - Traffic 2M bisa didapatkan **2 detik**



- Traffic 1.5M bisa didapatkan **4 detik**



Mengisi Bucket

- Setelah bucket kosong, bandwidth client dikunci pada token-rate (max-limit).
- Bucket akan terisi kembali ketika client tidak menggunakan bandwidth atau bandwidth real yang digunakan dibawah batas maksimal (max-limit).
- Konsep Token Bucket ini mirip seperti konsep menabung tanpa bunga ☺.

Efek Token Bucket

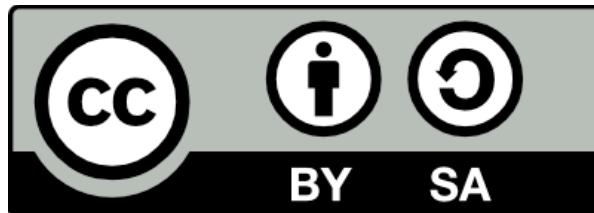
- Efek penggunaan Token Bucket ini jika diperhatikan mirip juga dengan fungsi Burst yang sudah ada
- Lalu mana yang lebih baik, Token Bucket atau Burst ????

Conclusion

- Ketika proses Token Bucket bisa menggantikan mekanisme Burst yang lama, diharapkan proses Bandwidth Management menjadi lebih ringan.
- Sangat berpengaruh terutama untuk perangkat Router Mikrotik yang memiliki processor mid range.

Thank you !!

www.mikrotik.co.id
info@mikrotik.co.id



- This license lets others remix, tweak, and build upon your work even for commercial purposes, as long as they credit you and license their new creations under the identical terms. This license is often compared to “copyleft” free and open source software licenses. All new works based on yours will carry the same license, so any derivatives will also allow commercial use.