

Berdasarkan dataset Anda yang memiliki fitur `policing_rate_kbps` dan `policing_burst_kbps` untuk setiap port, serta keinginan Anda untuk membandingkan model dengan ruang aksi diskrit (DQN, Dueling DQN) dan kontinu (PPO), berikut adalah parameter *action space* yang dapat Anda gunakan untuk mengatur *policing rate*:

1. Discrete Action Space (Untuk DQN & Dueling DQN)

Karena DQN dan Dueling DQN bekerja dengan ruang aksi diskrit, Anda tidak bisa meminta agen untuk langsung mengeluarkan nilai angka spesifik (misal: "set rate ke 5432 kbps"). Sebaliknya, Anda harus mendefinisikan aksi sebagai langkah penyesuaian bertahap. Berdasarkan pendekatan *step-based adjustment* yang umum digunakan dalam kontrol kemacetan 1, 2, berikut adalah definisi aksinya:

- **Jenis Aksi:** Penyesuaian Inkremental (Naik/Turun/Tetap).
- **Parameter Aksi (\mathbf{A}_t):** Agen memilih satu indeks integer yang dipetakan ke tindakan berikut untuk setiap Port (P1, P2, P4):
- **Action 0:** Maintain (Tidak ada perubahan pada `policing_rate`).
- **Action 1:** Increase Rate (Menambah `policing_rate_kbps` sebesar langkah tetap, misal $\Delta = 10\%$ atau 1 Mbps).
- **Action 2:** Decrease Rate (Mengurangi `policing_rate_kbps` sebesar Δ).
- **Action 3:** Increase Burst (Menambah `policing_burst_kbps` untuk mengakomodasi trafik mendadak).
- **Action 4:** Decrease Burst (Mengurangi `policing_burst_kbps`).

Implementasi pada 3 Port: Anda bisa menggunakan **Multi-Discrete Action Space** (jika library RL mendukung) atau menggabungkan kombinasi aksi jika menggunakan DQN standar (ini akan membuat ruang aksi sangat besar, jadi disarankan melatih agen terpisah per port atau menggunakan *Independent DQN*).

2. Continuous Action Space (Untuk PPO)

PPO sangat unggul di sini karena dapat mengeluarkan nilai *continuous* (angka riil) secara langsung, yang memungkinkan kontrol yang lebih halus dan presisi terhadap bandwidth 3, 4.

- **Jenis Aksi:** Direct Value Mapping atau Ratio Allocation.
- **Parameter Aksi (\mathbf{A}_t):** Agen mengeluarkan vektor nilai kontinu $\mathbf{a} \in \mathbb{R}^k$ (dimana k adalah jumlah parameter yang dikontrol).
- **Output 1 (Port 4 Rate):** Nilai $x \in [0, 1]$ yang dipetakan ke rentang $[\text{Min_Rate_P4}, \text{Max_Link_Capacity}]$.
- **Output 2 (Port 2 Rate):** Nilai $y \in [0, 1]$ yang dipetakan ke rentang bandwidth sisa.
- **Output 3 (Port 1 Rate):** Sisa bandwidth atau nilai independen.
- **Output 4-6 (Burst Sizes):** Nilai kontinu untuk mengatur `policing_burst_kbps` masing-masing port.

Keunggulan PPO: Seperti disebutkan dalam referensi, PPO dengan ruang aksi kontinu menghindari masalah "ledakan dimensi" yang terjadi pada DQN jika Anda ingin mengatur parameter secara presisi (misal: mengatur rate dengan granularitas 1 kbps) 6. PPO juga menggunakan mekanisme *clipping* untuk memastikan perubahan kebijakan (policy) tidak terlalu drastis, yang menjaga stabilitas jaringan 7, 8.

3. Parameter Teknis SDN untuk Eksekusi Aksi

Setelah agen RL (DQN/PPO) memilih aksi, "Action" tersebut harus diterjemahkan oleh *SDN Controller* (seperti Ryu) menjadi konfigurasi jaringan nyata. Berdasarkan referensi testbed SDN 9, 10, parameter teknis yang diubah adalah:

1. **Meter Bandwidth (Rate):** Mengubah parameter kbps pada *Meter Table* OpenFlow yang terpasang pada Port 1, 2, dan 4. Ini membatasi *throughput* maksimum.
2. **Burst Size:** Mengubah parameter *burst_size* pada *Meter Table*. Ini penting untuk Port 2 (Kamera) yang mungkin memiliki *frame I* besar sesekali, dan Port 4 (Healthcare) untuk mencegah *drop* paket saat data kritis dikirim mendadak.
3. **Queue Max-Rate:** Alternatif lain adalah mengatur max-rate pada antrian QoS (seperti Linux HTB) di level OVS (Open vSwitch) 2, 11.

Rekomendasi untuk Paper Anda: Dalam tabel perbandingan nanti, jelaskan bahwa:

- **DQN/Dueling DQN** menggunakan **Discrete Action** (Step \$10\%\$) yang mungkin menyebabkan osilasi atau konvergensi lambat ke nilai optimal yang presisi.
- **PPO** menggunakan **Continuous Action** yang memungkinkan penyesuaian *policing_rate* yang halus dan akurat, sehingga lebih efektif meminimalkan *SLA Violation* pada Port 4 (Healthcare).