

## 2.4 Parameter Performansi Protokol Komunikasi MQTT dan DDS

Pada penelitian ini, untuk mengukur kinerja protokol MQTT dan DDS, digunakan beberapa parameter sebagai berikut.

### 1. *Delay*

*Delay* adalah penundaan waktu suatu paket yang diakibatkan proses transmisi dari asal ke tujuan. Dalam pengujian tugas akhir ini berarti dari *publisher* yaitu sensor, menuju ke *subscriber* yaitu *client*. Cara menghitung *delay* dapat menggunakan persamaan beriku ini [16]:

$$\text{Delay} = \text{Waktu paket diterima} - \text{waktu paket dikirim} \quad (1)$$

### 2. *Drop Packet*

Adalah banyaknya paket yang hilang pada suatu jaringan paket yang disebabkan oleh tabrakan (*collision*), penuhnya kapasitas jaringan, dan penurunan paket yang disebabkan oleh habisnya TTL (*Time to Live*) dari paket [16]:

$$\text{Drop Paket} = \text{Jumlah paket dikirim} - \text{jumlah paket diterima} \quad (2)$$

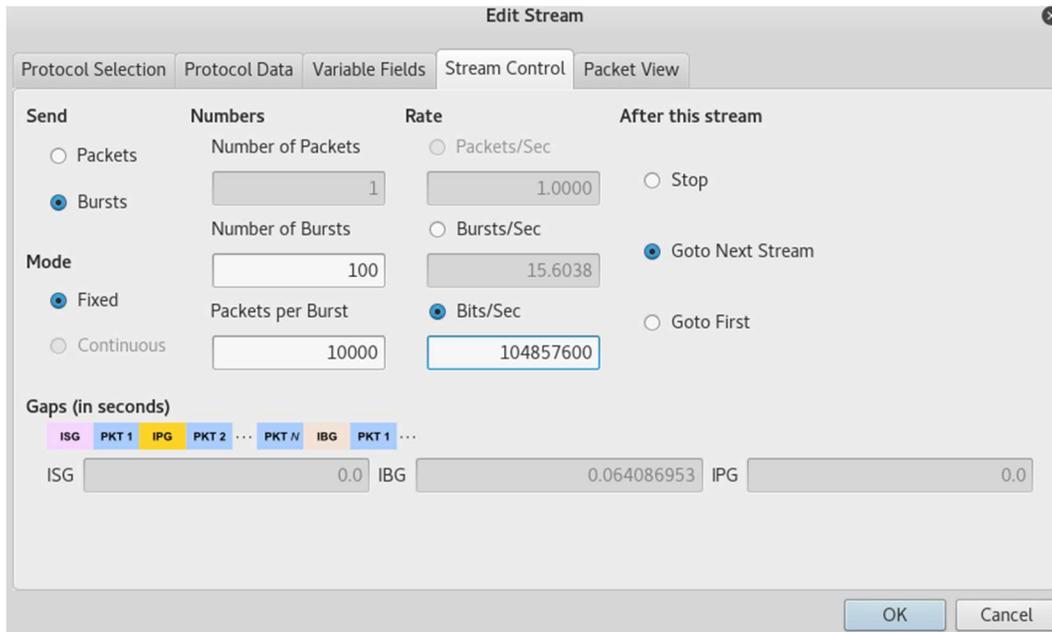
## 2.5 Traffic Generator

*Traffic Generator* adalah aplikasi yang dapat mengemulasi paket untuk lalu lintas jaringan. Salah satu aplikasi semacam ini adalah Ostinato, yang dapat mengemulasi dan mengirimkan sejumlah paket secara berurut maupun secara *burst* pada protokol yang berbeda [12].

Pada penelitian ini *traffic background* harus dapat memenuhi spesifikasi kabel UTP Cat5, yaitu mentransmisi data sebesar 100 Mbps. Sehingga pada konfigurasi Ostinato paket *burst* dikonfigurasi sebagai berikut:

- *Traffic background* yang digunakan adalah UDP.
- *Number of burst* adalah 100, artinya Ostinato mengirim sebanyak 100 kali paket, dan jeda mengirim setiap paketnya yaitu 5 sec, sesuai pilihan protokol yang digunakan, yaitu UDP.

- Pada *packets per burst* dikonfigurasi 100, 10.000, dan 1.000.000.
- Untuk Bits/sec karena harus memenuhi spesifikasi kabel UTP Cat5 100 Mbps, maka pada Bits/sec memiliki nilai 104857600 sesuai dengan konversi 100Mbps ke Bit/sec.

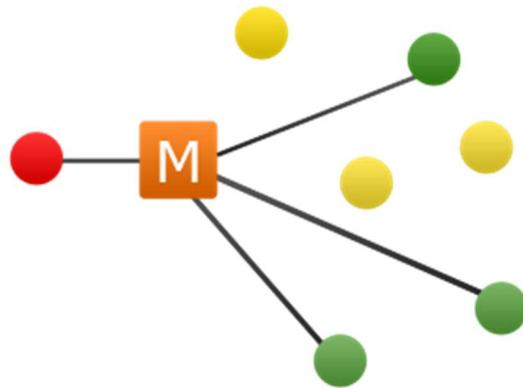


Gambar 2.5 Konfigurasi Ostinato untuk Paket Burst

## 2.6 Komunikasi Data *One-to-Many* Secara *Multicast*

*Multicast* atau *multicasting* adalah sebuah teknik di mana sebuah data dikirimkan melalui jaringan ke sekumpulan komputer yang tergabung ke dalam sebuah grup tertentu. Pada penelitian ini adalah sebuah data sensor dengan satu topik dikirimkan oleh satu *publisher* melalui jaringan protokol MQTT maupun DDS menuju group *subscriber*, yaitu aplikasi yang menerima data sensor tersebut.

*Multicast* merupakan sebuah cara pentransmisi data secara *connectionless* (komunikasi dapat terjadi tanpa adanya negosiasi pembuatan koneksi), dan klien (*subscriber*) dapat menerima transmisi *multicast* dengan mencari topik apa yang sama antara *publisher* dan *subscriber*. *Multicast* sebenarnya merupakan mekanisme *one-to-many*, atau *point-to-multipoint*.



Gambar 2.6 Skenario *Multicast*

Pada arsitektur jaringan menggunakan layanan *multicast*, memungkinkan apabila pengiriman satu data dari satu *publisher* dapat langsung terkirim secara bersamaan menuju seluruh *subscriber*. Hal tersebut memungkinkan pengiriman data dengan cepat dengan penggunaan *bandwidth* yang efisien [16].

## 3 Perancangan Sistem

### 3.1 Gambaran Sistem

Pada tahap ini dilakukan perancangan terhadap sistem yang akan dibuat sesuai skenario agar permasalahan dapat terselesaikan. Hasil perancangan akan dijabarkan dengan mendefinisikan kebutuhan fungsional berupa komponen perangkat lunak dan keras yang akan digunakan, dan alur sistem.

#### 3.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam perancangan sistem adalah sbb:

##### 1. Komputer *Server*

Sebagai *server* yang nantinya berfungsi sebagai broker dan hanya digunakan untuk implementasi protokol MQTT. Namun broker MQTT pada studi ini akan dipindahkan pada infrastruktur AWS (*Amazon Web Service*) yang merupakan layanan infrastruktur berbasis *cloud*. Adapun spesifikasi dari *server* tersebut adalah sbb:

- a. *Processor* : Intel® Xeon Processor
- b. *vCPU* : 1.00 GB
- c. *RAM* : 1.00 GB
- d. *Storage* : *EBS-Only (Elastic Block Store)*
- e. *OS* : Ubuntu 16.04

##### 2. Komputer Virtual Sensor (*publisher*)

Berfungsi sebagai sensor virtual dan mengirimkan data *dummy* ke aplikasi melalui broker. Virtual sensor ini berfungsi sebagai *publisher* baik pada infrastruktur untuk layanan MQTT maupun DDS. Spesifikasi perangkat keras yang digunakan sebagai *publisher* ini adalah sbb:

- a. *Processor* : Intel® Core i5-3370 CPU @ 1.80GHz
- b. *RAM* : 4.00GB
- c. *Harddisk* : 500GB
- d. *OS* : Kali Linux