**Background**
- Emergence of various wireless communication standards.
- Cases in which computers have several Network-Interfaces (N/I) have increased ex) 802.11abgn, Bluetooth, UWB, PHP-network, 2G or 3G cellular phone etc...
- People want to use the interfaces at the same time and make the best use of bandwidth, ex) web surfing, mail transfer, multimedia streaming etc...

We propose A Socket-level Bandwidth Aggregation Mechanism SBAM.

**Approach**

**Socket-layer**
- Reuse of existing software without rewrite.
- End-to-End nature
- Scalability to N/I.
- Transport-protocol-independent.

**Link/Network Layer**
- It is impossible to exchange data between end-hosts because of Hop-by-Hop nature.
- Deploying anything except TCP/UDP is difficult.
- It's necessary to rewrite existing softwares.
- The scalability against the number of NICs is low.

**Design**

**Policy Transfer Function**
- Exchanges of user policy through MIB between kernel space and user space.

**Network Monitoring Function**
- Monitoring of delay and bandwidth of each link.

**Send Data Scheduling Function**
- Resolution of amount of data to send according to the Network condition.

**Send Data Division Function**
- Passing the data to lower protocol according to MTU and amount of data.

**Incoming Data Aggregation Function**
- Uniting the information which was divided and sent, and pass it to application.

**Policy for sending data**

Send Data Scheduling Function

\[ P_n = \frac{\alpha \cdot b_n \cdot d_n}{\min(\alpha \cdot b_1 \cdot d_1, \ldots, \alpha \cdot b_n \cdot d_n)} (0 \leq \alpha \leq 1) \]

\[ P_n = \frac{\alpha \cdot b_n \cdot \text{lcm}(m_1, \ldots, m_n)}{m_n \cdot \gcd(h_{b_1}, \ldots, h_{b_n})} \]

\[ P_n: \text{Num of pkts} \quad \alpha_n: \text{Link Weight} \quad m_n: \text{MTU size(bytes)} \quad d_n: \text{Delay(ms)} \quad b_n: \text{Bandwidth(bps)} \]

Network Monitoring
- Delay: Measures by sending ICMP packet regularly.
- \[ \text{SRTT} = \alpha \text{SRTT} + (1 - \alpha) \times \text{RTT} \quad (0 \leq \alpha \leq 1) \]
- Bandwidth: Uses packet-pair technique.
- \[ \text{Bandwidth} = \frac{\text{Packet size}}{\text{Diff of RTT}} \]