A Path-Aware Scheduling Scheme for MPTCP

Draft-zuo-mptcp-scheduler-01

J. Zuo, J. Zhu, W. Liu
Huawei

IETF-101, MPTCP WG, March 2018

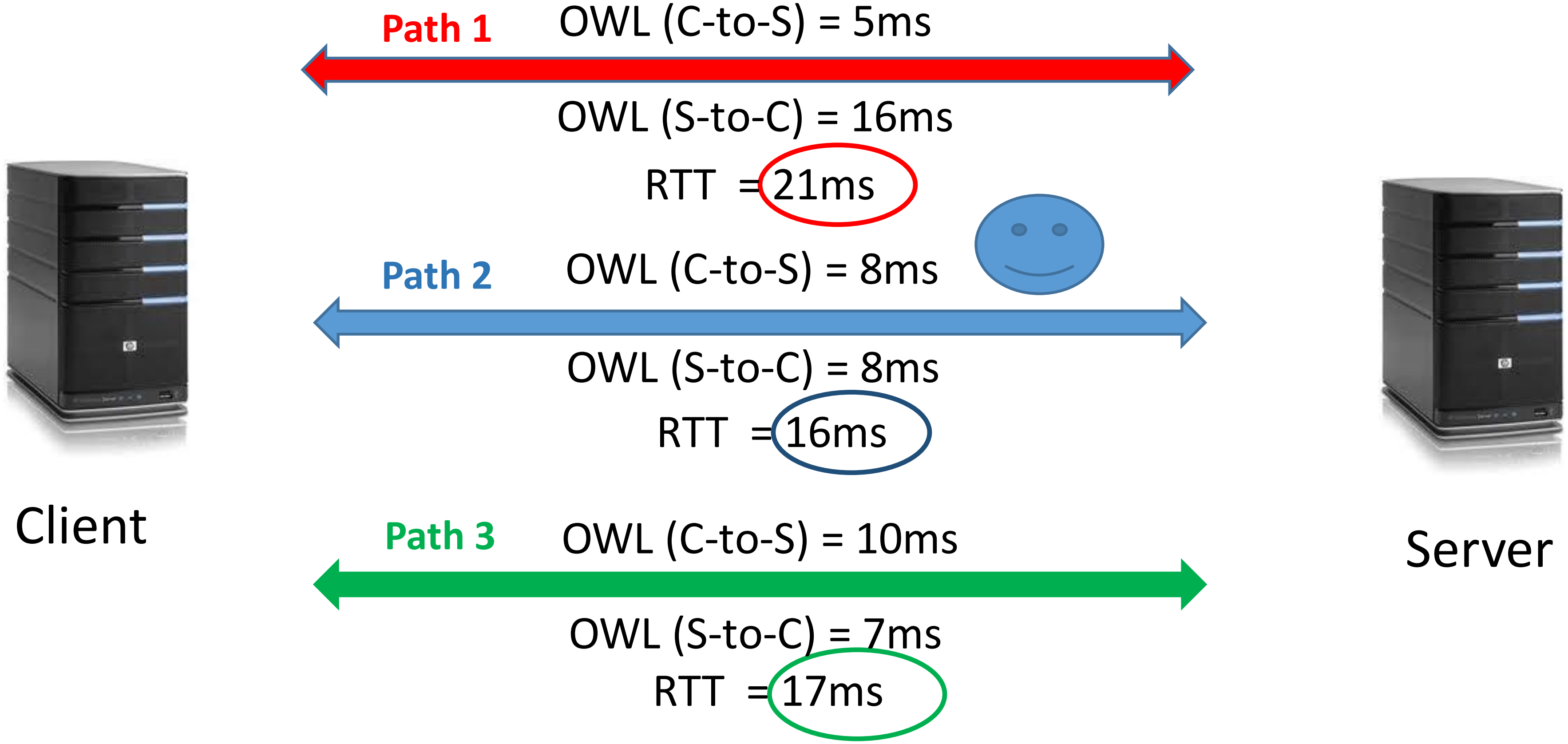
Latency-Sensitive applications

- Latency-Sensitive applications require the data delivered in the path with the lowest latency

- RTT is commonly used as a condition for data scheduling amongst multiple paths
- However, the delays of the forward path and the reverse path may be different in
  - ➢ wireless environment
  - ➢ scenario when congestion causes the different queue delays

- Better to consider OWL for data scheduling

Example: Scheduling based on RTT



Client

Server

Path 1

OWL (C-to-S) = 5ms

OWL (S-to-C) = 16ms

RTT = 21ms

Path 2

OWL (C-to-S) = 8ms

OWL (S-to-C) = 8ms

RTT = 16ms

Path 3

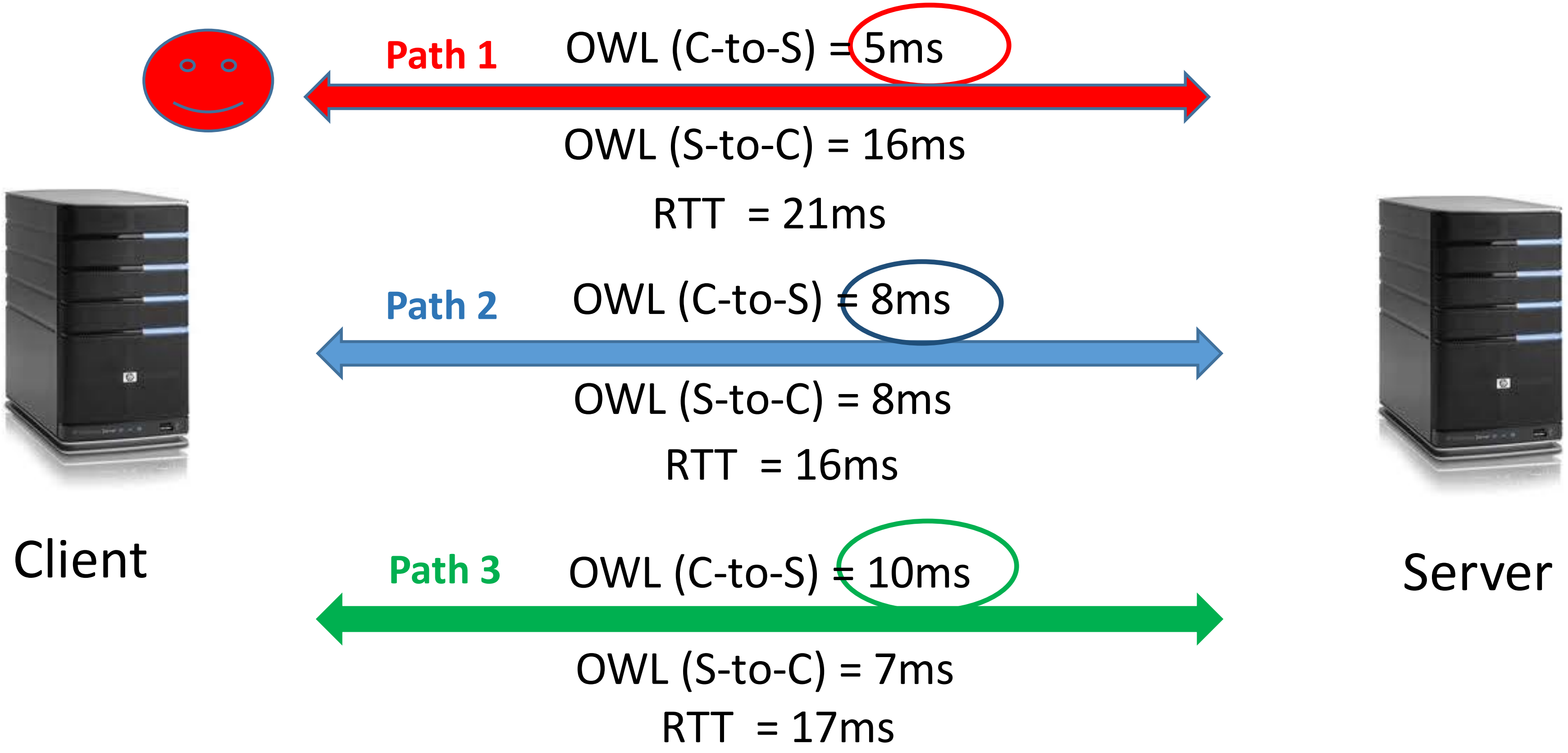
OWL (C-to-S) = 10ms

OWL (S-to-C) = 7ms

RTT = 17ms

Data will be scheduled in Path 2 based on RTT.

Example: Scheduling based on OWL



Path 1 OWL (C-to-S) = 5ms

OWL (S-to-C) = 16ms

RTT = 21ms

Path 2 OWL (C-to-S) = 8ms

OWL (S-to-C) = 8ms

RTT = 16ms

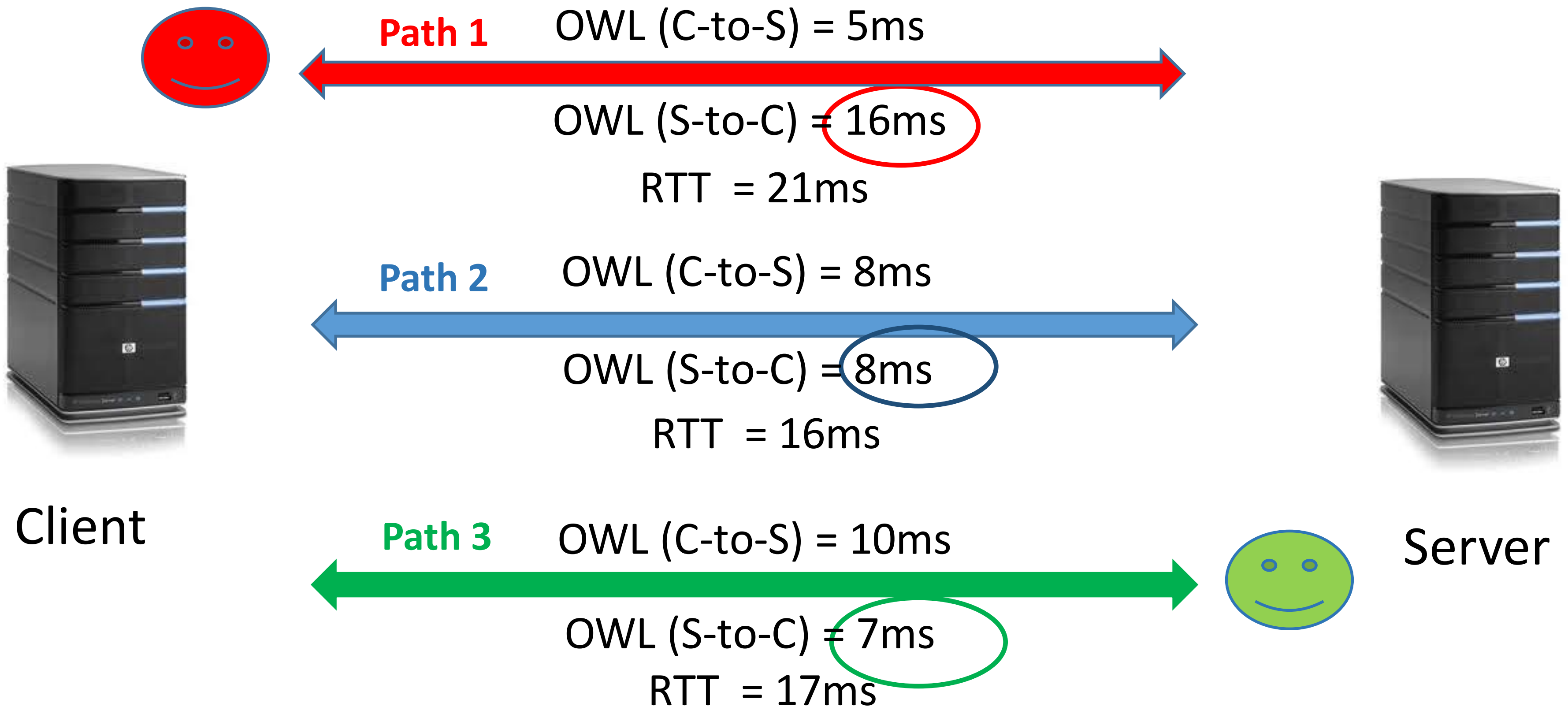
Path 3 OWL (C-to-S) = 10ms

OWL (S-to-C) = 7ms

RTT = 17ms

Data will be scheduled in Path 1 based on OWL.

Example: Scheduling based on OWL



Path 1 OWL (C-to-S) = 5ms

OWL (S-to-C) = 16ms

RTT = 21ms

Path 2 OWL (C-to-S) = 8ms

OWL (S-to-C) = 8ms

RTT = 16ms

Path 3 OWL (C-to-S) = 10ms

OWL (S-to-C) = 7ms

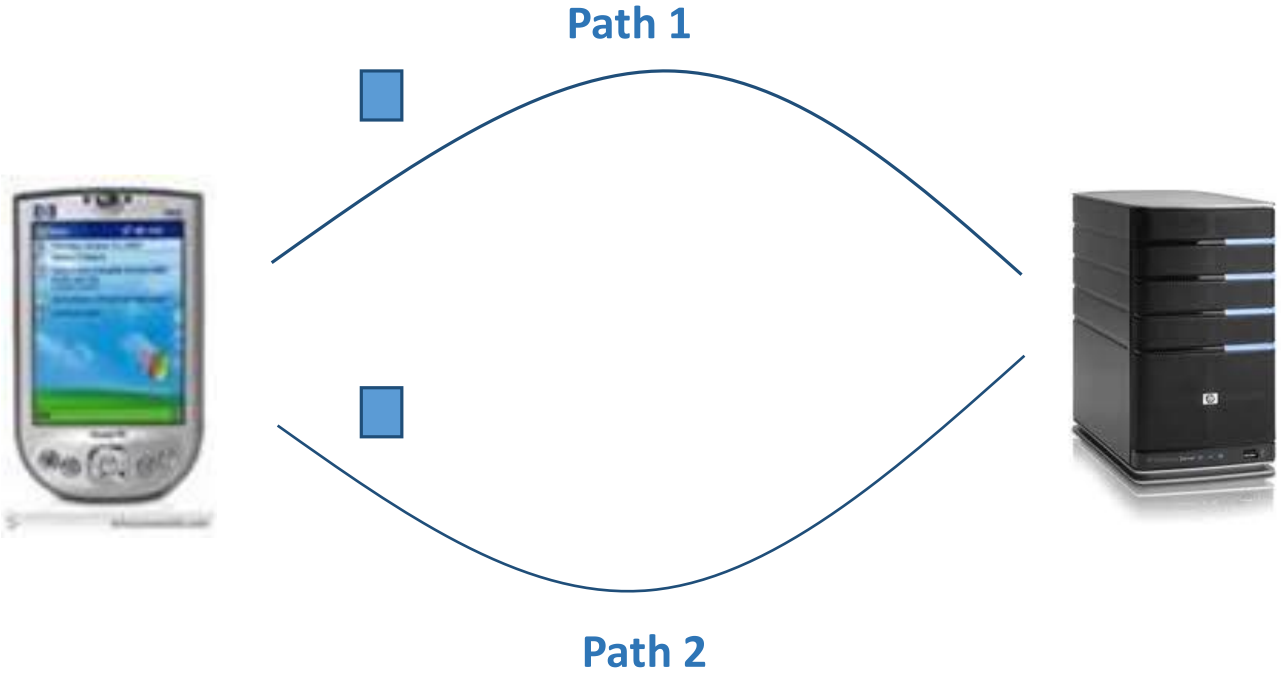
RTT = 17ms

Client

Server

Considering the interactive latency,
data will be scheduled in Path 1, while ACK will scheduled in Path 3.

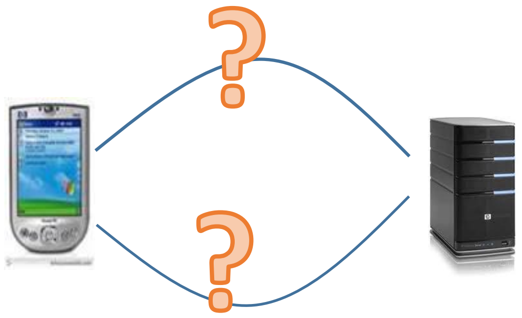
A Path-Aware Scheduling Scheme



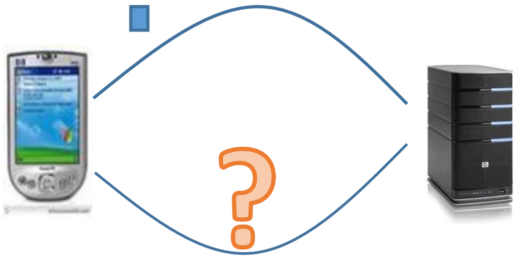
Design principles:

- employ redundant transmission when path characteristics are unknown;
- always send data in the path with the lowest OWL;
- periodically update the OWLs of all paths and schedule data again.

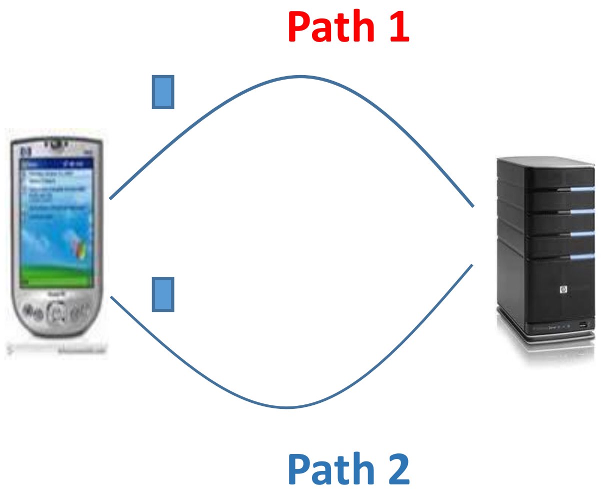
Initialization



unknown path characteristics



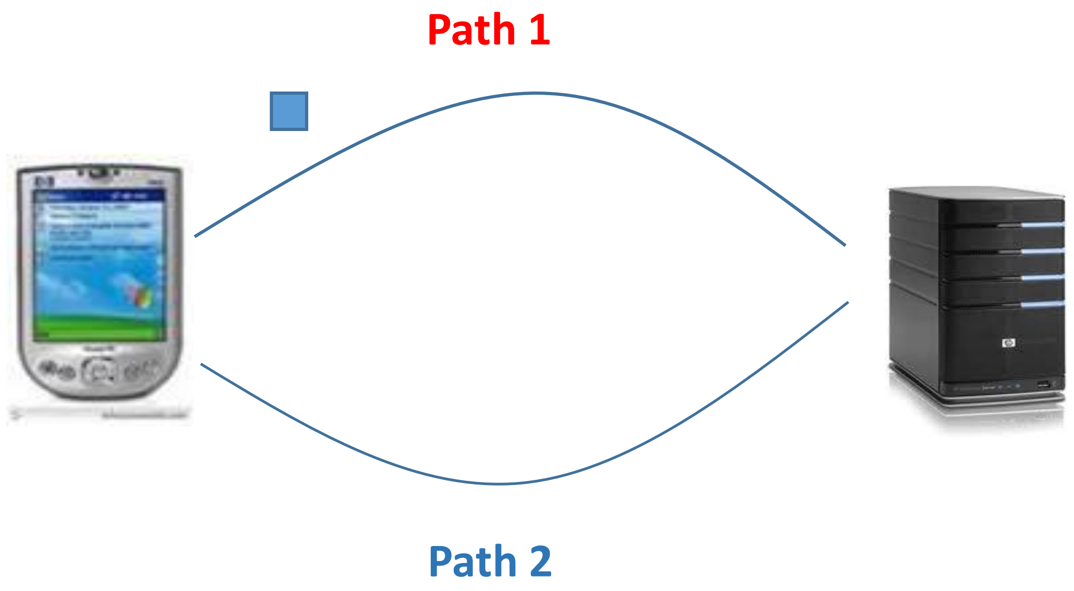
one path connected



redundant transmission

- transmit data redundantly until obtain the effective OWLs of all paths;
- / transmit data redundantly for a period of time (e.g. 1s).

Packet Scheduling



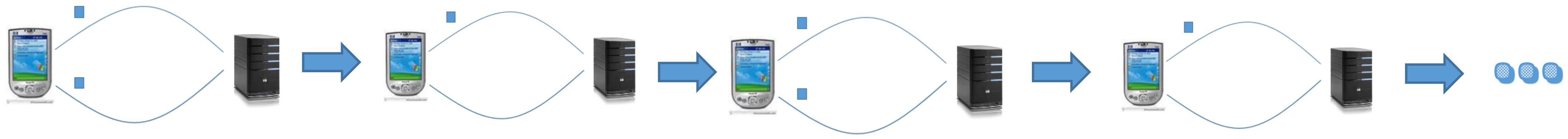
OWL (1) = T_recv (1) - T_send (1) + dT
OWL (2) = T_recv (2) - T_send (2) + dT
where
T_send (i) is the sending time of the data;
T_recv(i) is the receiving time of the data;
dT is the the time difference caused by the absolute clock time.

dOWL = OWL (1) - OWL (2)

- no time synchronization issue

- Always send data in the path with the lowest OWL, for example,
  If dOWL < 0, send data in Path 1;
  Else send data in Path 2.

Periodically Redundant Transmission



1s

Redundant transmission

10s

Transmission in the path with the lowest OWL

1s

Redundant transmission

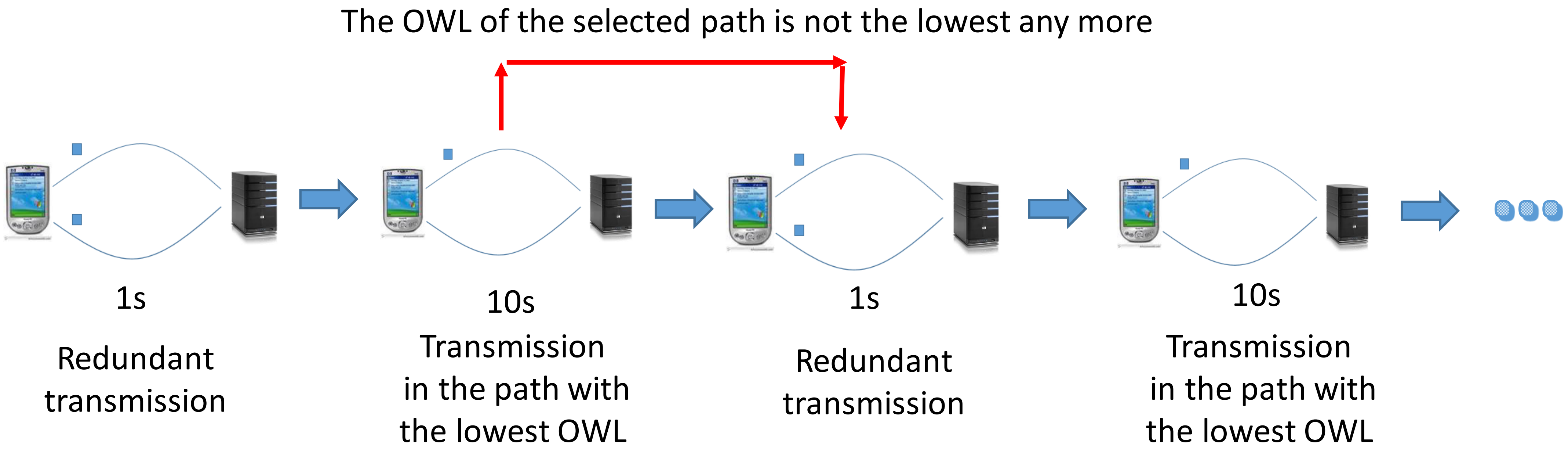
10s

Transmission in the path with the lowest OWL

Every 10 seconds, redundant transmission is activated, which

- obtain the OWLs of all paths in the same time;
- without introducing extra packets;
- guarantee the lowest delivery time no matter which path the data goes through.

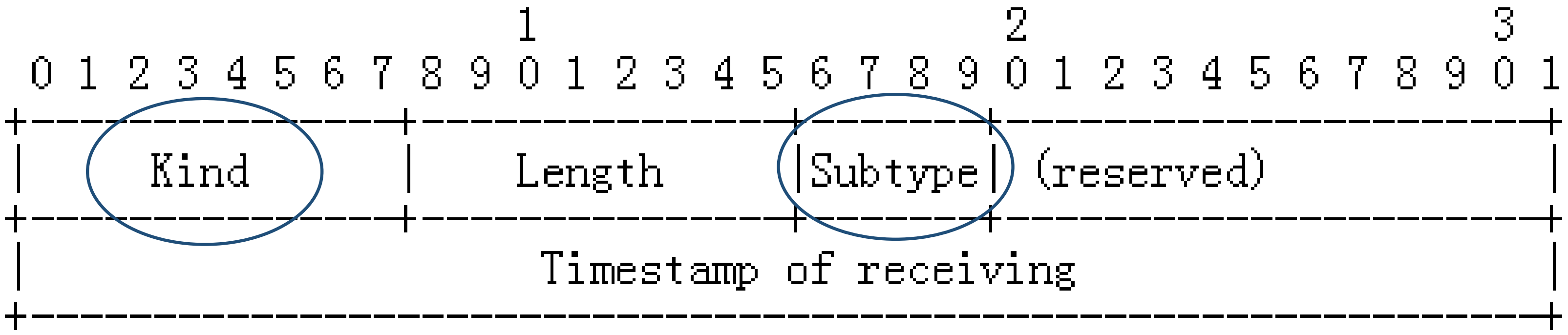
Immediately Activate the Redundant Transmission


The OWL of the selected path is not the lowest any more
1s
Redundant transmission
10s
Transmission in the path with the lowest OWL
1s
Redundant transmission
10s
Transmission in the path with the lowest OWL

- When the OWL of the selected path has increased so much that it may not be the lowest any more, the redundant transmission is activated immediately.

Implementation Consideration

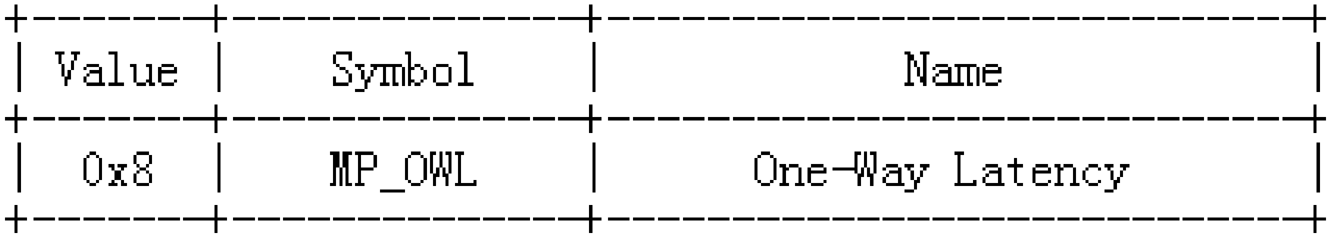
- A new MPTCP option (MP_OWL Option) is defined to carry the timestamp of data receiving at the receiver.



```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-------+-----------------------+
|     Kind      |    Length     |Subtype| (reserved)            |
+---------------+---------------+-------+-----------------------+
|                   Timestamp of receiving                      |
+---------------------------------------------------------------+
```

MP_OWL option (Kind = 30)

[RFC6824]



Value	Symbol	Name
0x8	MP_OWL	One-Way Latency

The subtype of MP_OWL option (Subtype = 0x8)

[RFC6824][RFC5226]

OWL Calculation

- Negotiation needed ensuring MP_OWL option is supported

- The steps of OWL calculation:

  - ➢ the sender sends each data and remember T_send of the data;
  - ➢ the receiver responses an ACK with T_recv of the data;
  - ➢ the sender fetches T_recv from the ACK and subtracts T_send of the data to get the OWL (OWL = T_recv - T_send).

Thank you

Questions/Comments

Any interest in continuing this work?