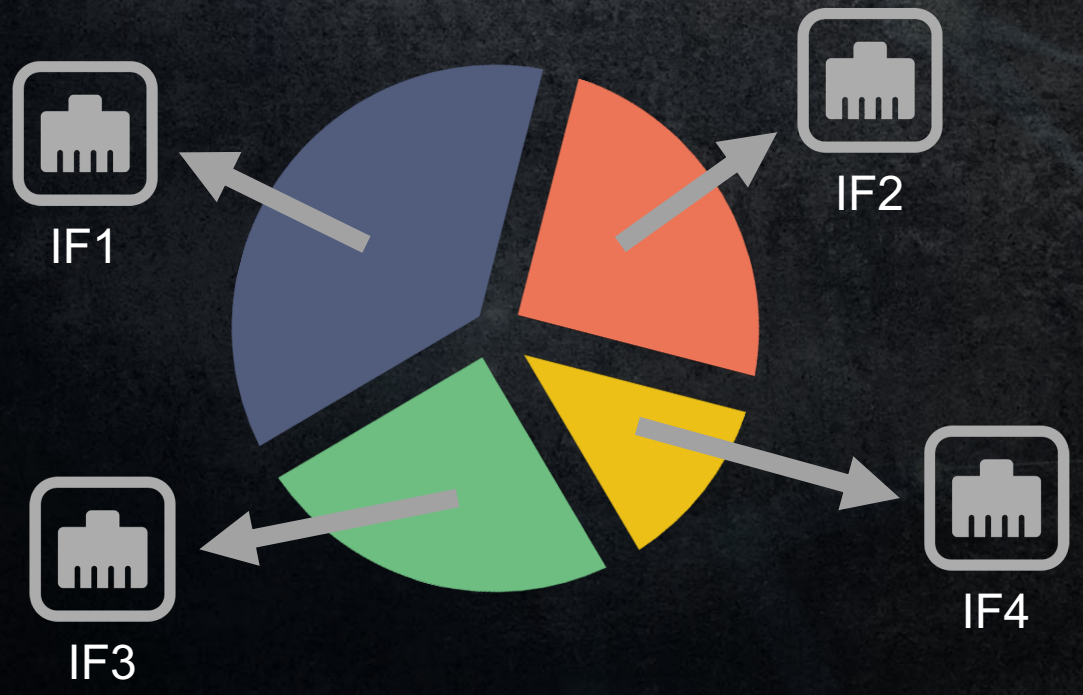
agile content

Bonding

EFP Plug-in

BONDING PLUG-IN

- Distributing EFP fragments over multiple interfaces
- Supports 1+n configurations
- Supports {n...} array of interfaces
  - Where the interface data coverage is defined >0% to <=100%



SAME SIMPLE C++ TYPE API'S ELASTICFRAMEPROTOCOL HAS

Create your EFPBonding plug-in

```
//EFP Bonding plug-in
EFPBonding myEFPBonding;
```

Create all interfaces

```
//Build new interfaces
groupInterfacesID[0] = myEFPBonding.generateInterfaceID();
lInterfaces.push_back(EFPBonding::EFPInterface(std::bind(&networkInterface1, std::placeholders::_1),
                                               groupInterfacesID[0],
                                               EFP_MASTER_INTERFACE));
groupInterfacesID[1] = myEFPBonding.generateInterfaceID();
lInterfaces.push_back(EFPBonding::EFPInterface(std::bind(&networkInterface2, std::placeholders::_1),
                                               groupInterfacesID[1],
                                               EFP_NORMAL_INTERFACE));
```

Create bonding group

```
groupID[0] = myEFPBonding.addInterfaceGroup(lInterfaces);
```

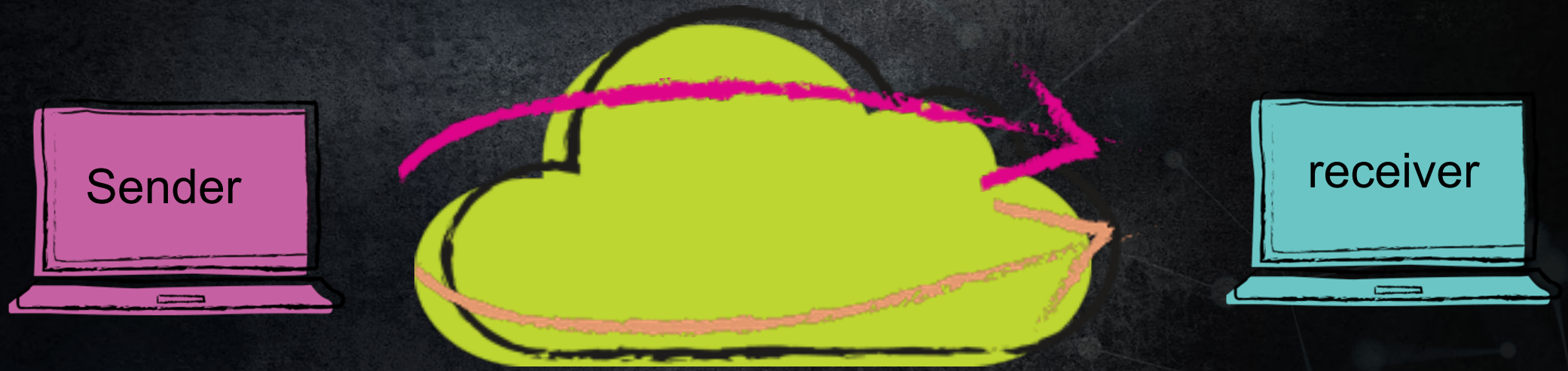
Integrate with EFP

```
void sendData(const std::vector<uint8_t> &rSubPacket, uint8_t fragmentID) {
  myEFPBonding.distributeDataGroup(rSubPacket, 0);
```

DONE!!

USECASES

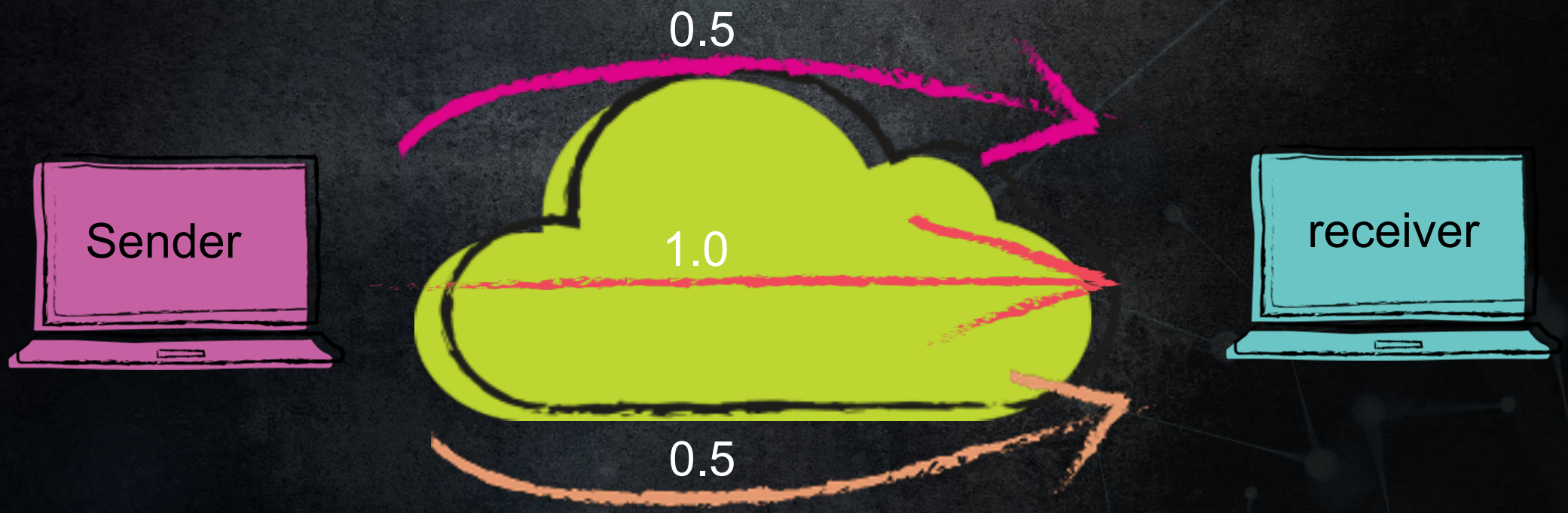
1 + 1



Use case: Protect your data

CAN ALSO DO FRACTIONAL

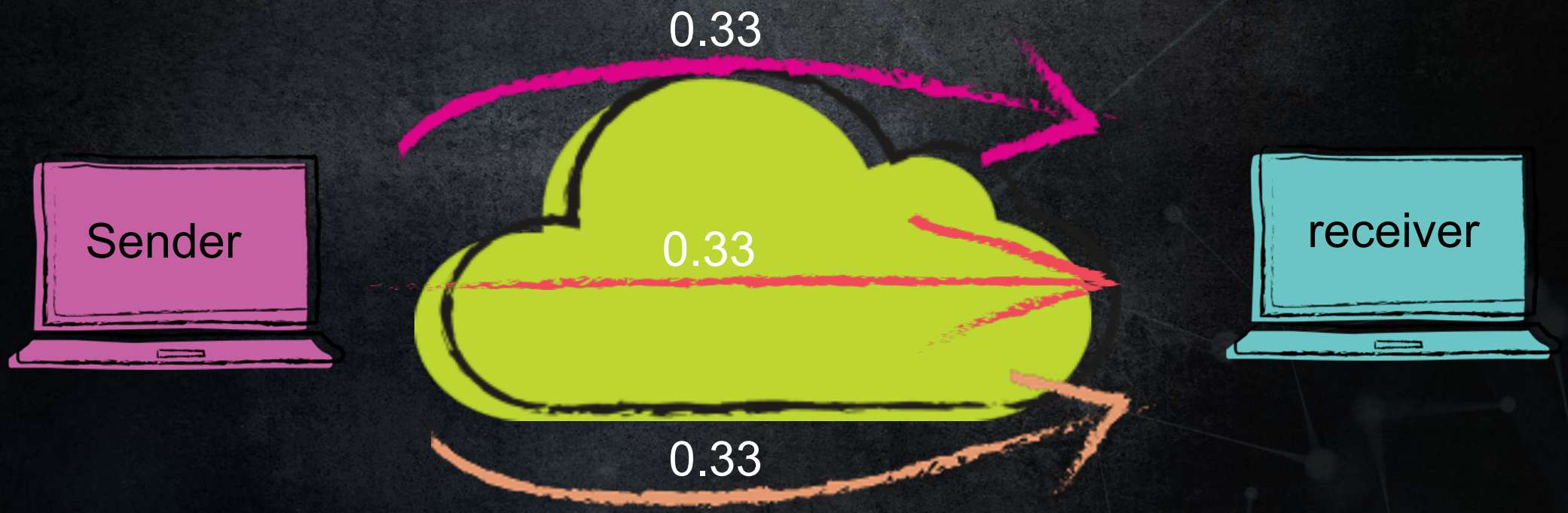
For example 1 + 0.5 + 0.5



Use case: Protect your data but not all links can carry 100% payload

BONDING

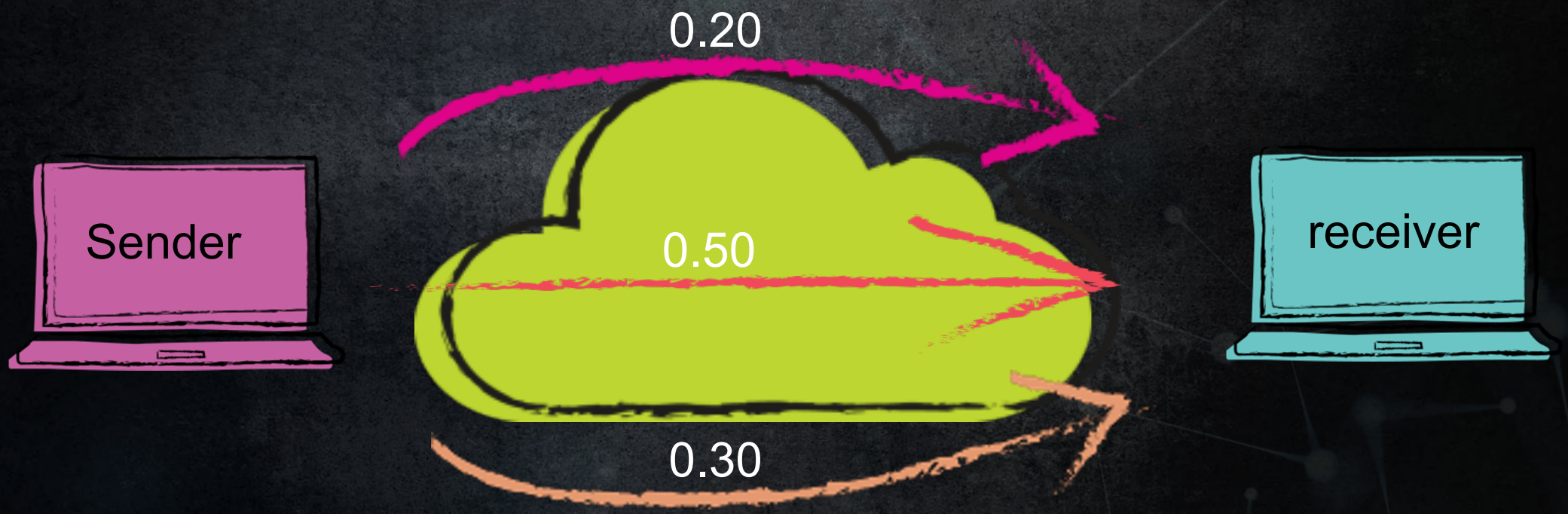
For example 0.33 + 0.33 + 0.33



Use case: Send data over 4G links or spread
100% capacity over multiple interfaces

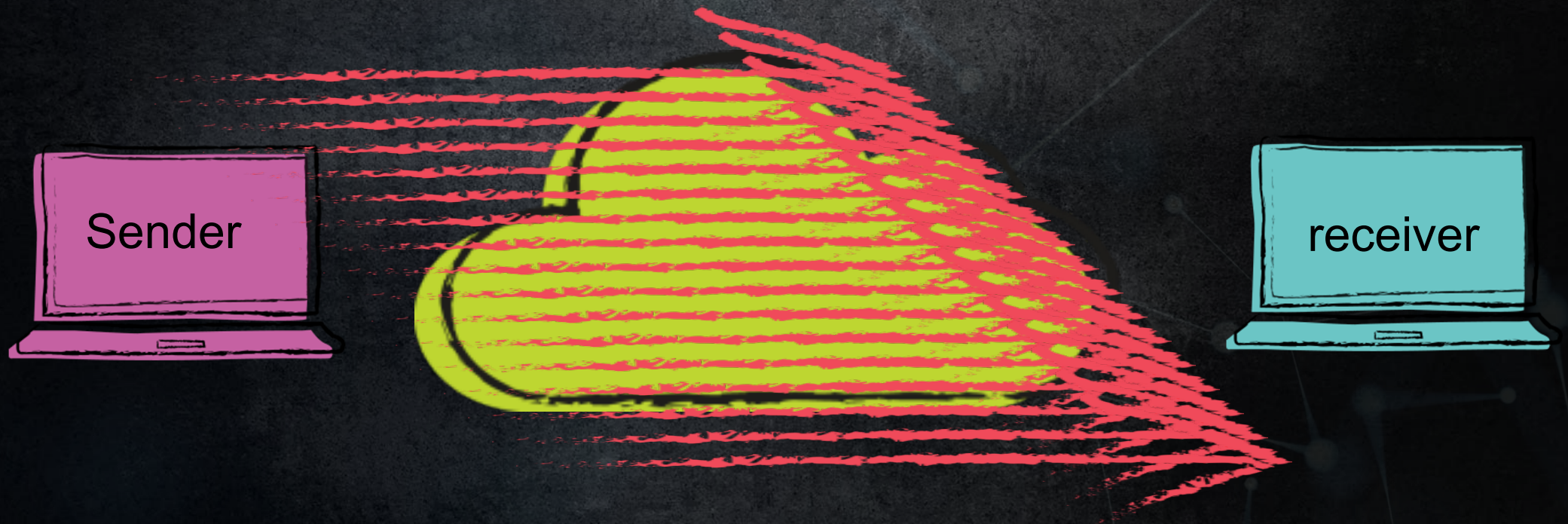
BONDING

For example 0.33 + 0.33 + 0.33



Use case: Send data over multiple links to their capacity

DATA DISTRIBUTION OVER ANY NUMBER INTERFACES



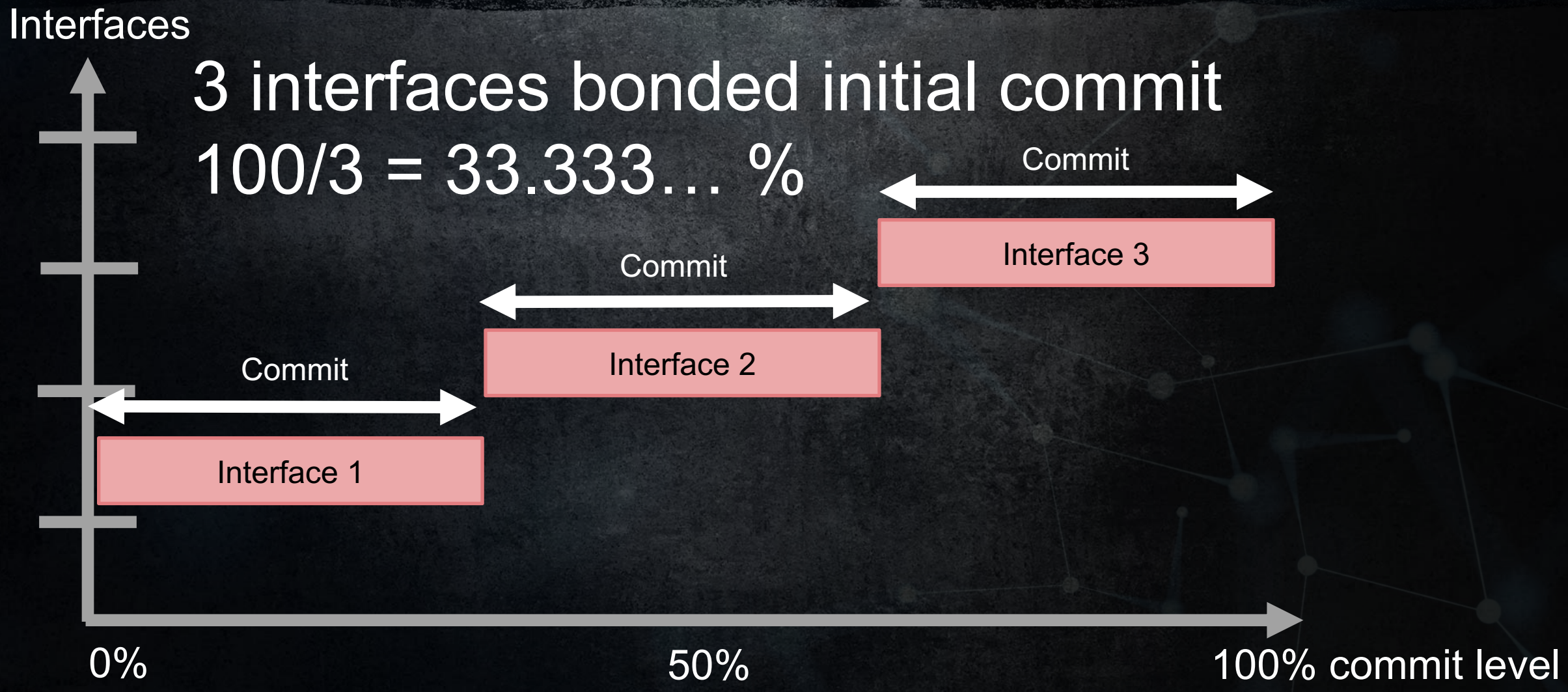
Any interface/link commit to a certain % payload of the total generated BW (100%)

BONDING

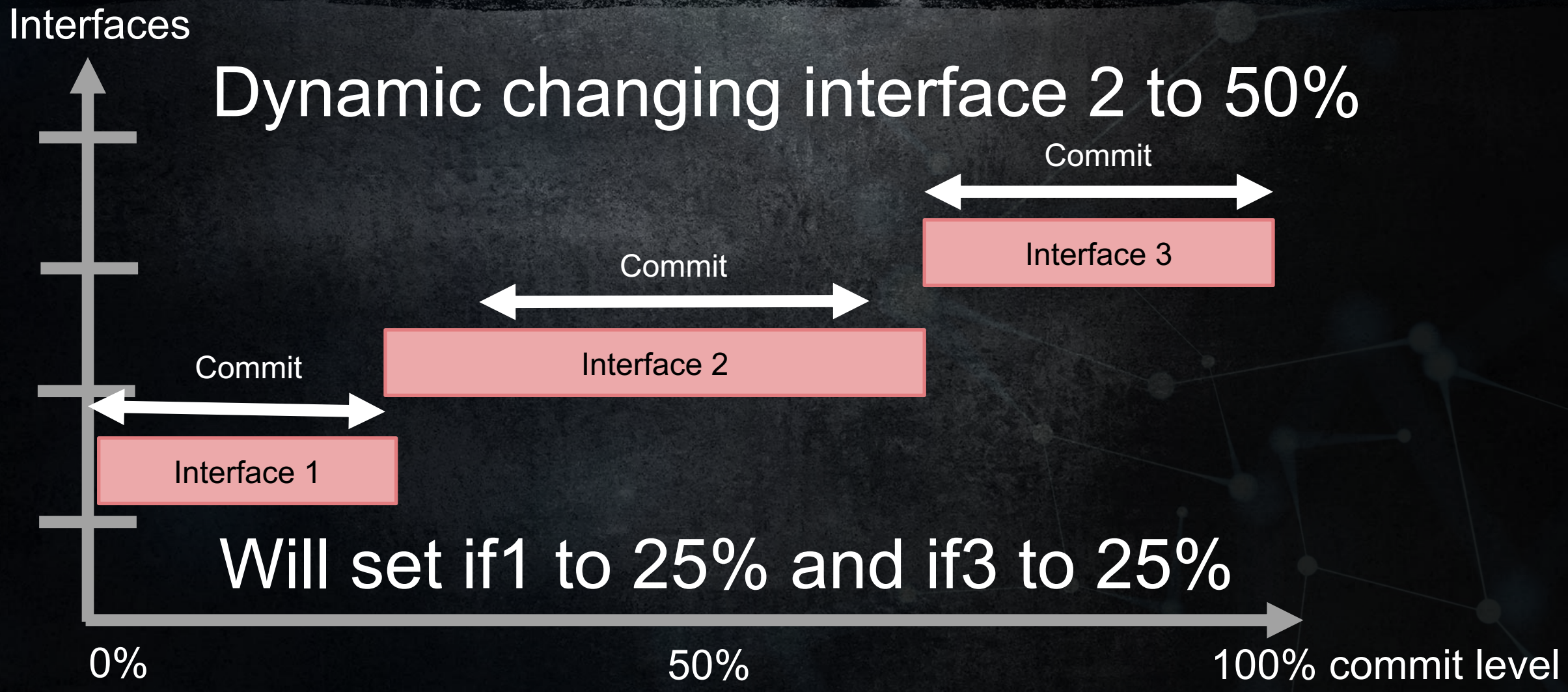
EFP Plug-in

DETAILS

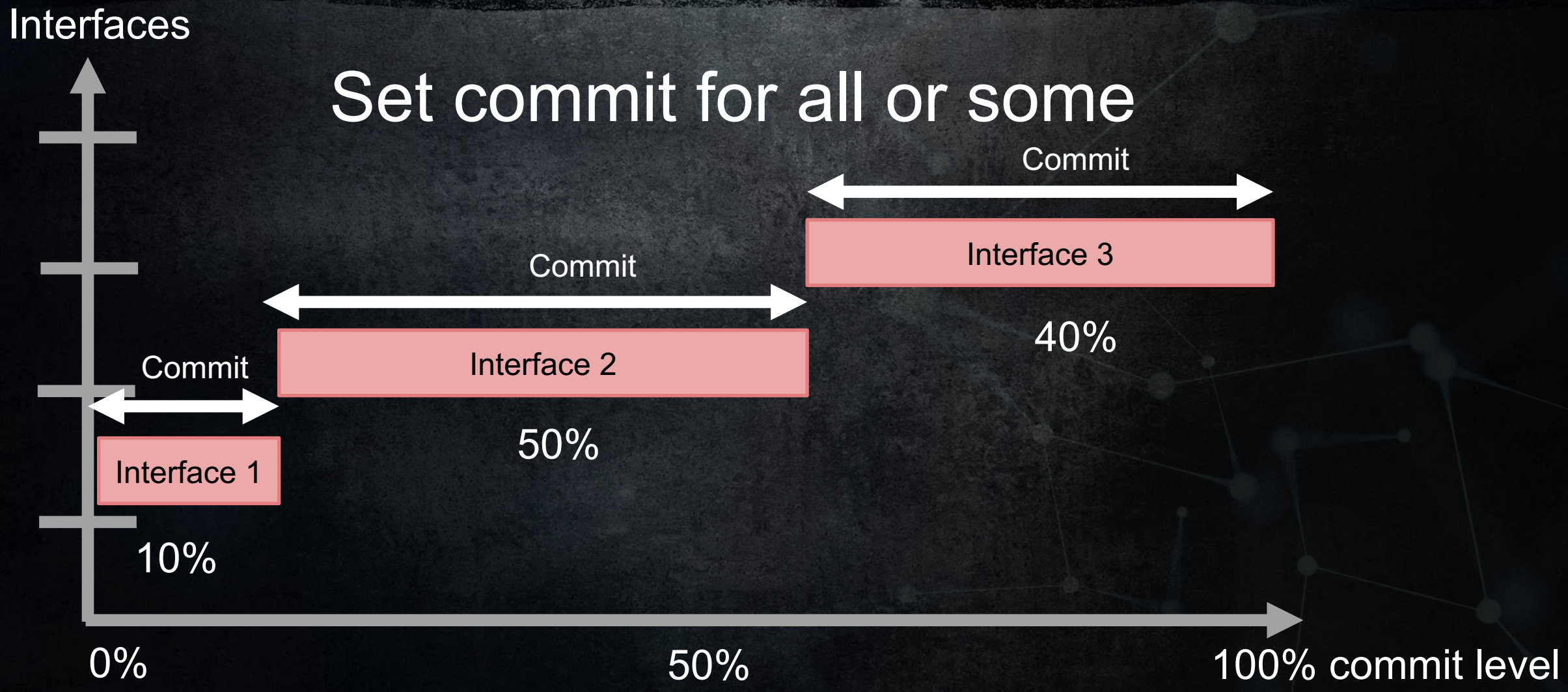
CONFIGURATION OF THE INTERFACE



CONFIGURATION OF THE INTERFACE



CONFIGURATION OF THE INTERFACE



CONFIGURATION OF THE INTERFACE

- Re-configuring interfaces can be made while operating
- Remember to call from same thread as ’ distributeDataGroup’
- Change individual interfaces (Other interfaces in the group will cover up to 100% commit or give up capacity if the interface wants to commit to more than it previously did)
- Change the whole group.

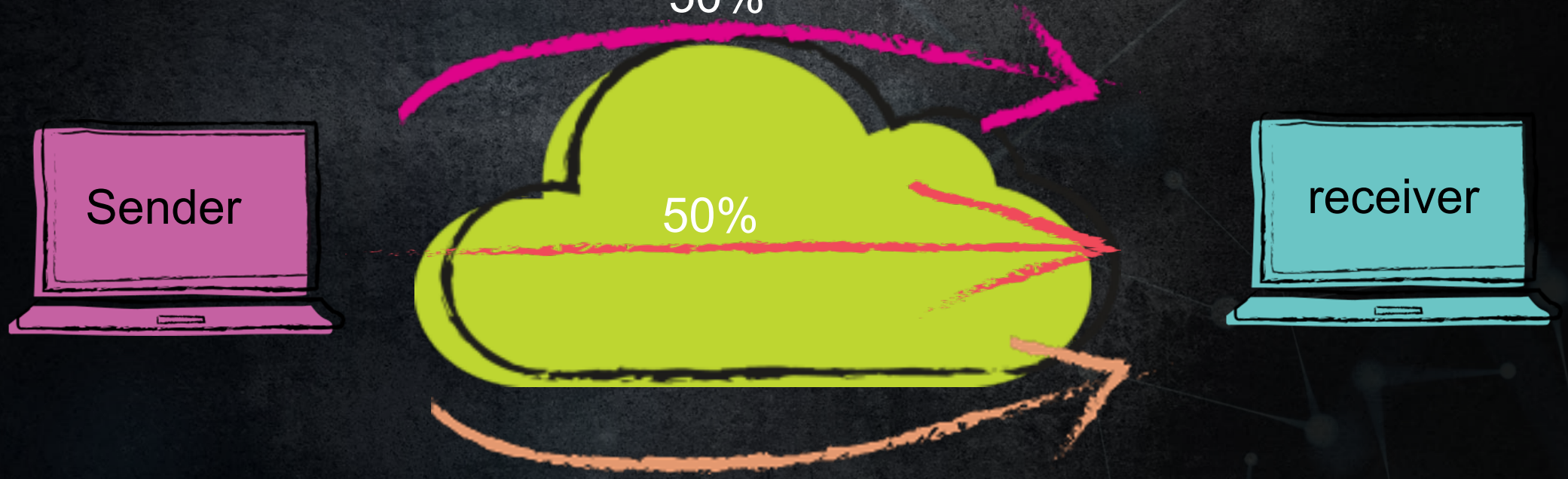
SPLIT-MODE

SPLIT – MODE BASICS

- An interface can belong to a split-mode group.
- Split mode is splitting the traffic from EFP based on EFP-StremID.
- An Interface can belong to a split-mode group and/or a load balance group
- If an EFP-Stream is undefined in the split mode group the fragment can be dropped or sent to the master interface.

SPLIT – MODE USE CASE

Load balance 50% + 50%



Split mode EFP StreamID = 5

In this case we use 2 x 4G modems to increase our BW. We got a third interface of some sort where we send only the audio. This link is reliable meaning if we were to drop any of the 4G links the receiver would still get the audio

SPLIT – MODE USE CASE

1+1 stream ID 5

Split mode EFP StreamID = 5

Split mode EFP StreamID = 5

Simple 1+1 but only Stream ID 5 all other streams are dropped

SPLIT – MODE USE CASE

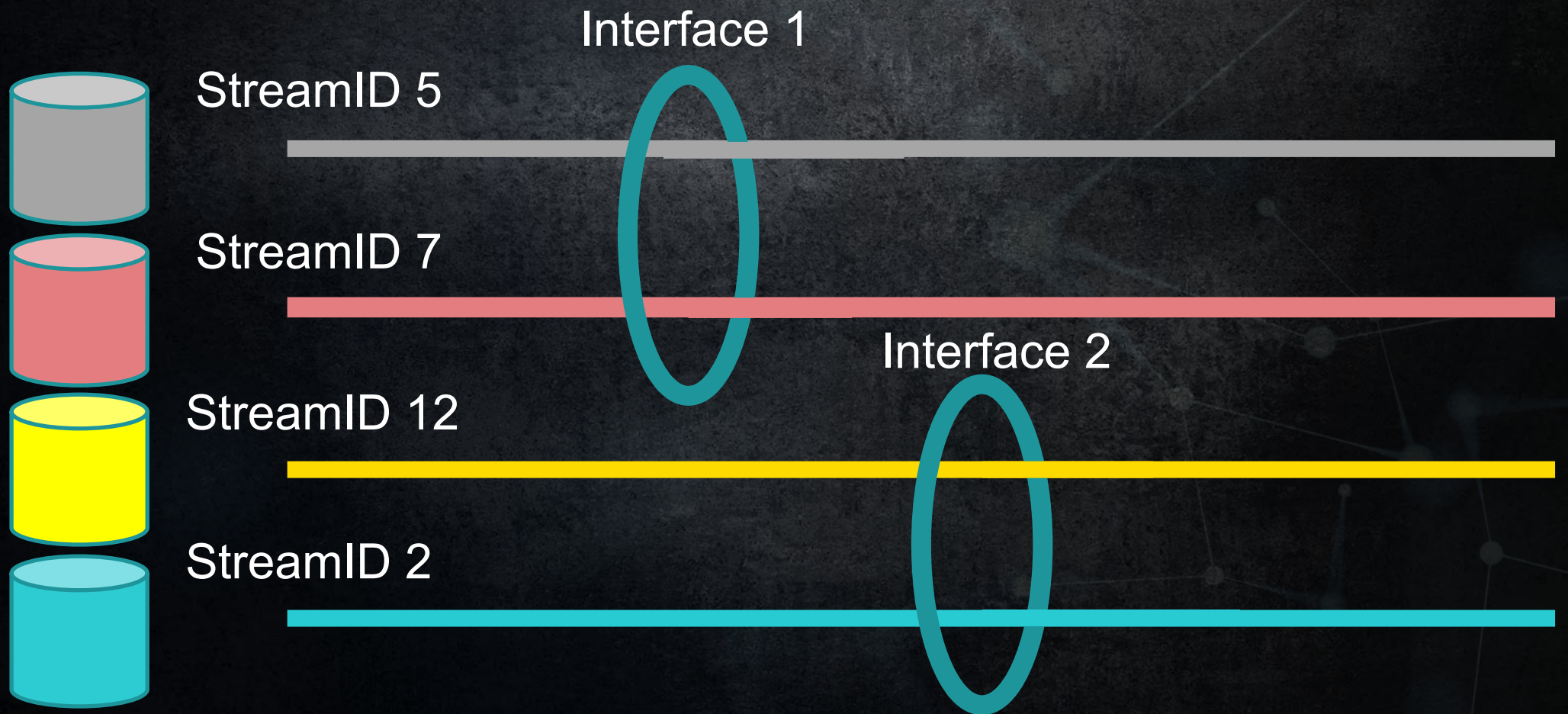
1+1 stream ID 5

Split mode EFP StreamID = 5

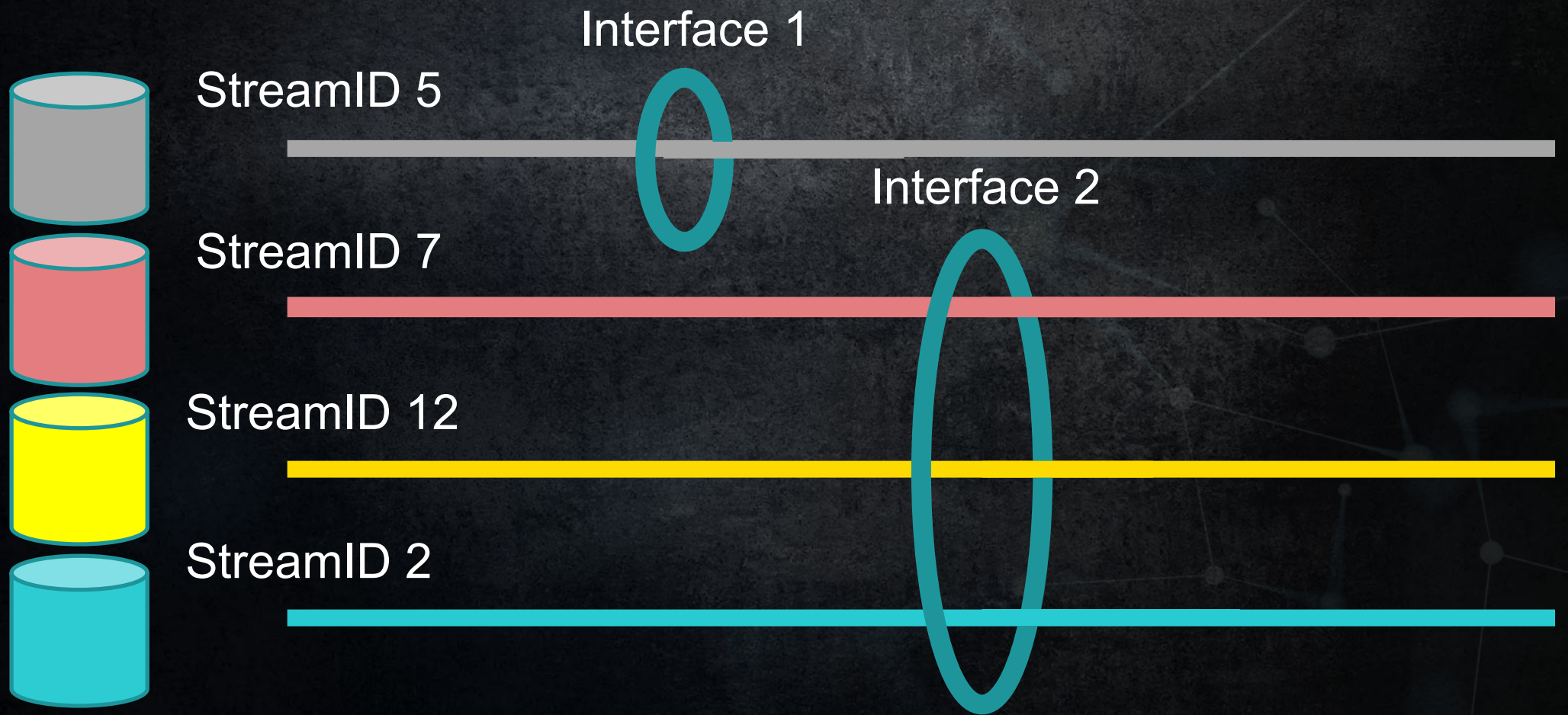
Split mode EFP StreamID = 5,4,12,33

1+1 for ID 5, the other streams are sent on one interface

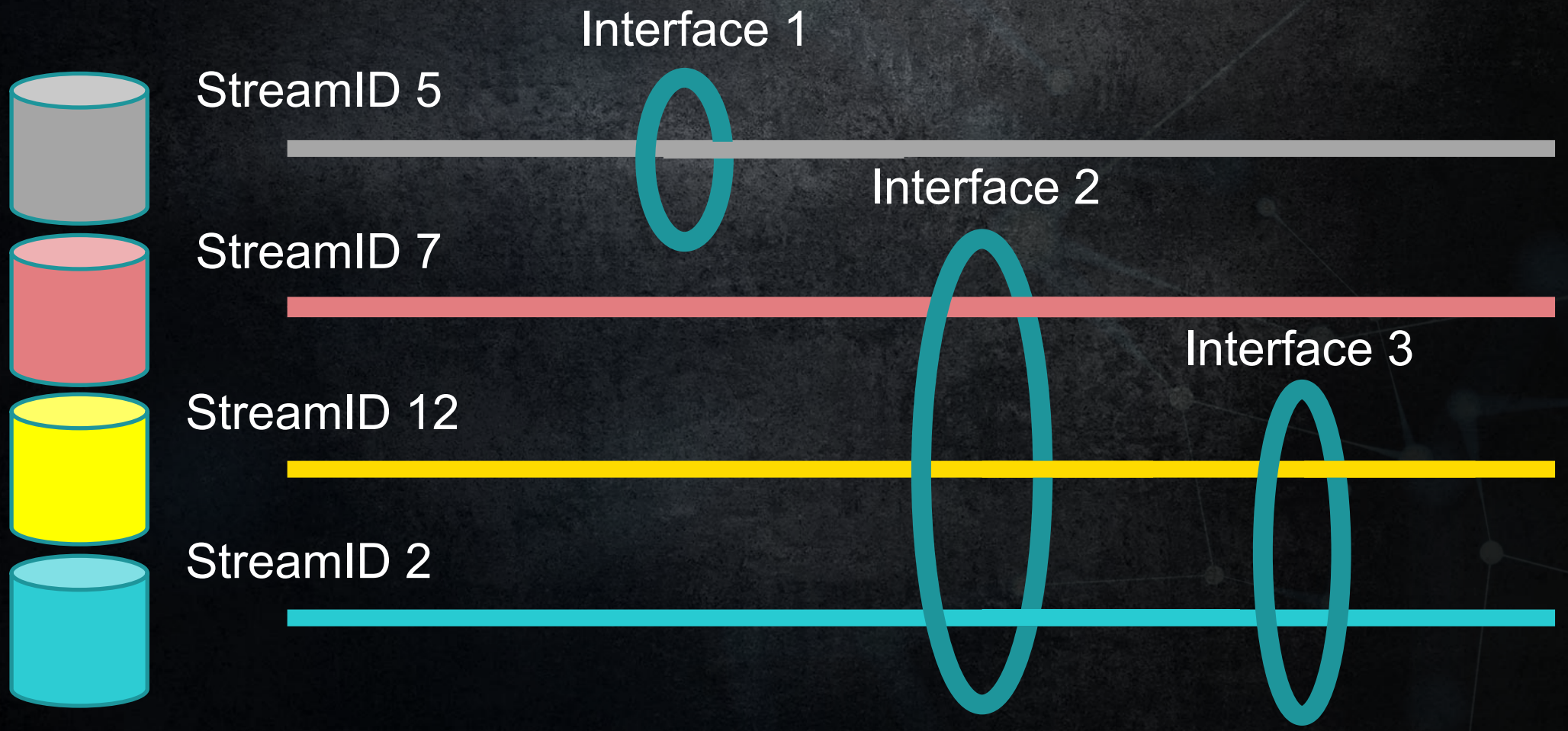
SPLIT MODE ILLUSTRATED (EXAMPLE)



SPLIT MODE ILLUSTRATED (EXAMPLE)



SPLIT MODE ILLUSTRATED (EXAMPLE)



STATISTICS

STATISTICS

Get Statistics ->

```
EFPBonding::EFPStatistics thisInterfaceStatistics = myEFPBonding.getStatistics(groupInterfacesID[x], groupID[0], true);
std::cout << "If: " << unsigned(x) <<
          " fragments: " << unsigned(thisInterfaceStatistics.mNoFragmentsSent) <<
          " part: " << thisInterfaceStatistics.mPercentOfTotalTraffic << "%" <<
          " cover fragments: " << unsigned(thisInterfaceStatistics.mNoGapsCoveredFor) <<
          std::endl;
```

```
class EFPStatistics {
public:
  uint64_t noGapsCoveredFor = 0;
  uint64_t noFragmentsSent = 0;
  double percentOfTotalTraffic = 0;
};
```

If this is a master interface I cover for fractional calculation packets.
No packets covered for

Fragments sent over this interface

What is the % load on this interface

STATISTICS

Using ->

```
class EFPStatistics {
public:
  uint64_t noGapsCoveredFor = 0;
  uint64_t noFragmentsSent = 0;
  double percentOfTotalTraffic = 0;
};
```

and ->

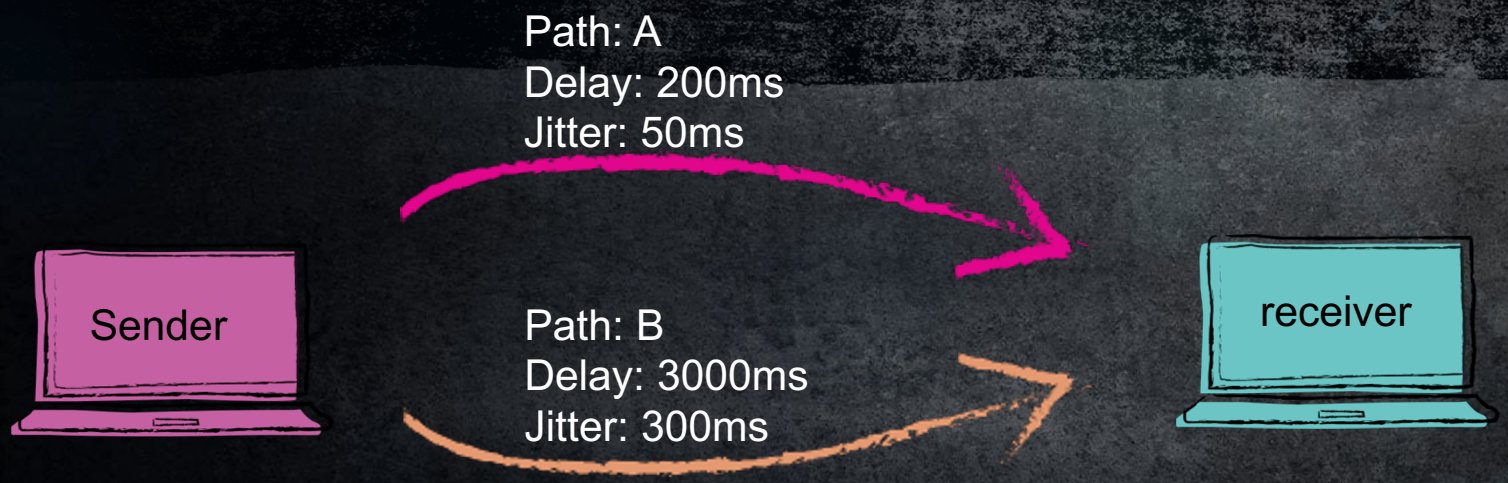
```
uint64_t EFPBonding::getGlobalPacketCounter() {
  return globalPacketCounter;
}
```

You can ->

- Bandwidth per interface and second
- PPS
- + more

CONFIGURE EFP CORRECT

CONFIGURE EFP



Calculation->
Worst delta case Path A Low 150ms Path B high 3300ms = delta 3150ms
If the protocol under EFP is coping with out of order packets, then -> myEFPReciever (3150, 0.....
If the underlying protocol can handle out of order data leave the ´0´, however if not then change the ´0´ to number of ms to wait for out of ordered frames before timing out the not complete super frames.

agile content

ElasticFrame Protocol

Using

BONDING EFP Plug-in

THANKS FOR ATTENDING