- 本資料は下記論文を基にして作成されたものです．文書の内容の正確さは保障できないため，正確な知識を求める方は原文を参照してください．

- 題目：Integrating Security, Mobility, and Multi-homing in a HIP Way
- 著者：Pekka Nikander, Jukka Ylitalo, and Jorma Wall
- 発行：2003.02
- 発行所：Ericsson Research Nomadic Lab.

HIPによるセキュリティ・モビリティ・マルチホーミングの統合

名城大学 理工学部
渡邊研究室
050427125 水谷智大

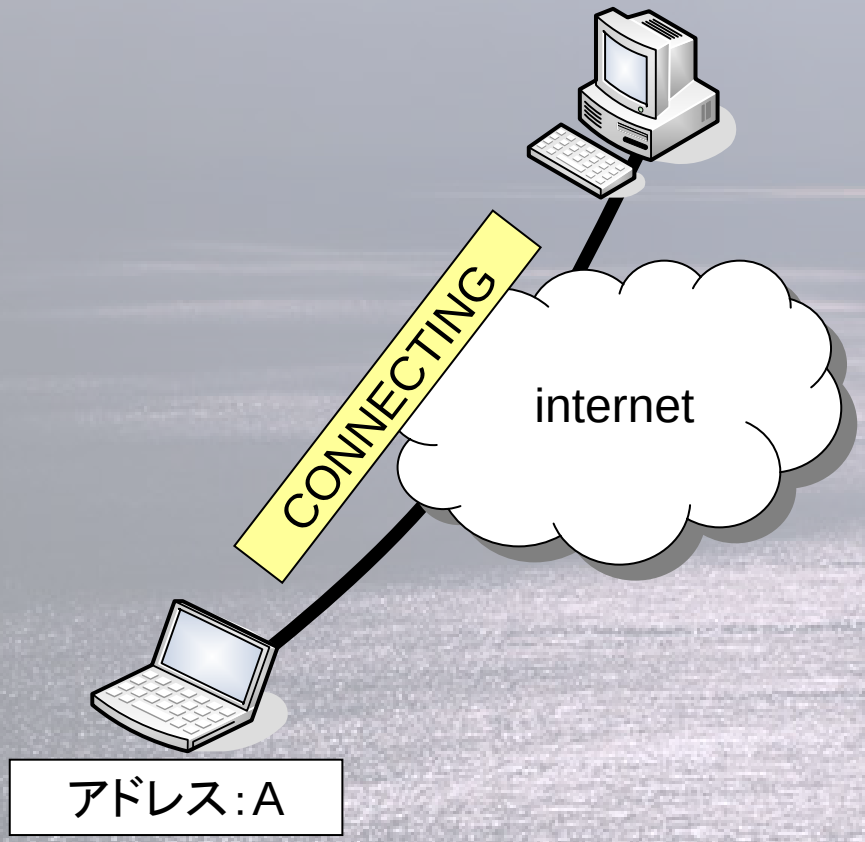
研究背景

- 近年のモビリティ・マルチホーミングに対する需要の高まり
  - しかし，端末の移動にはアドレスの変化が避けられない
  - マルチホーミングの実現には専門知識や特殊機器が必要

解決

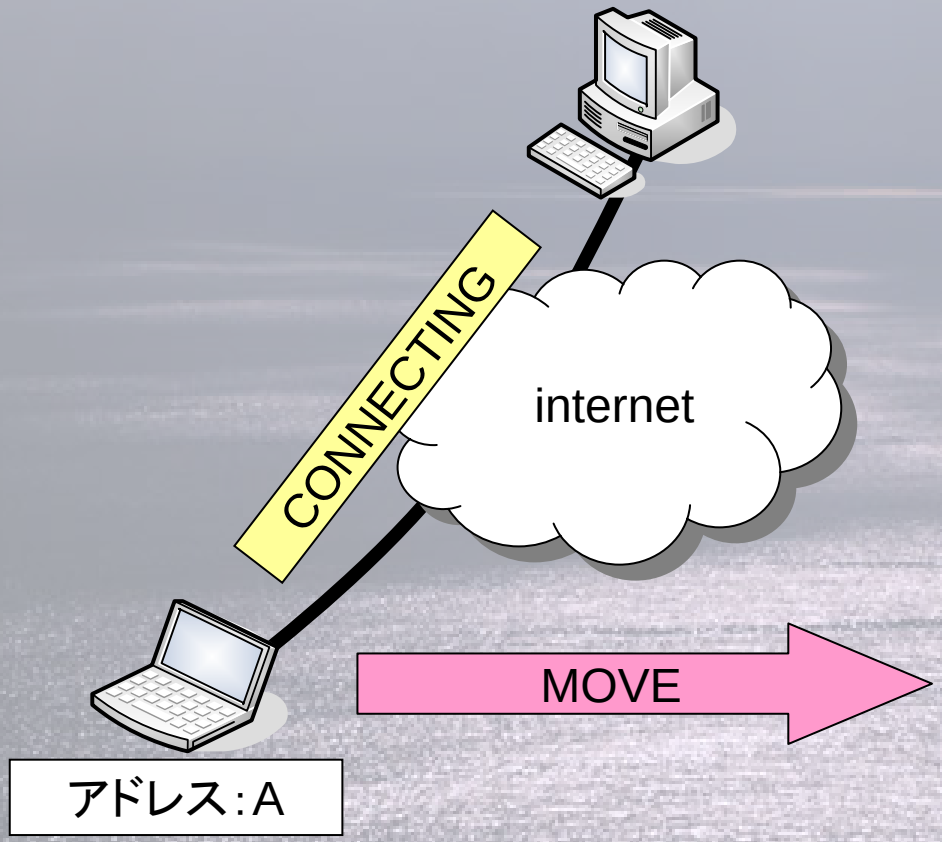
実現

モビリティ



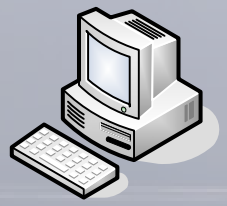
通信を継続したまま移動することが可能

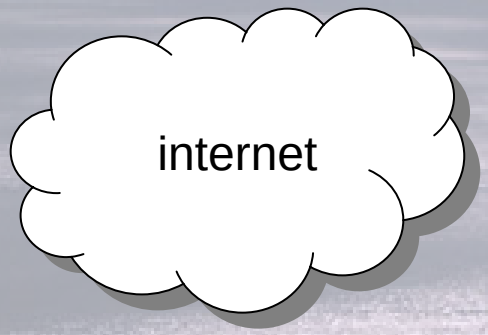
モビリティ

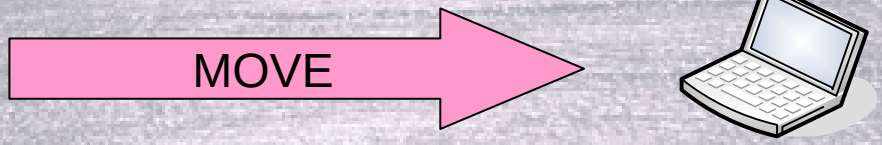


通信を継続したまま移動することが可能

モビリティ



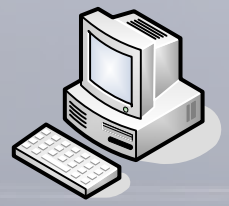


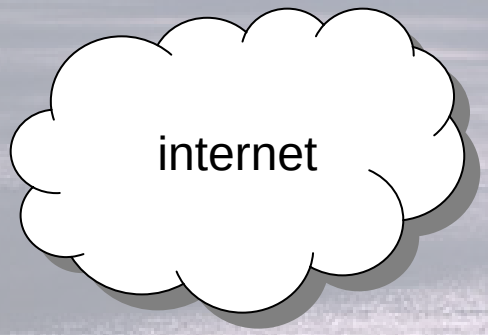


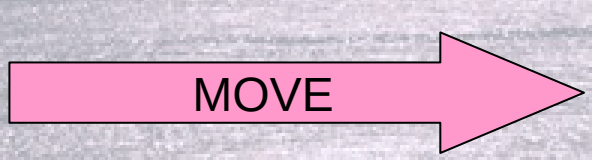
アドレス：A

通信を継続したまま移動することが可能

モビリティ



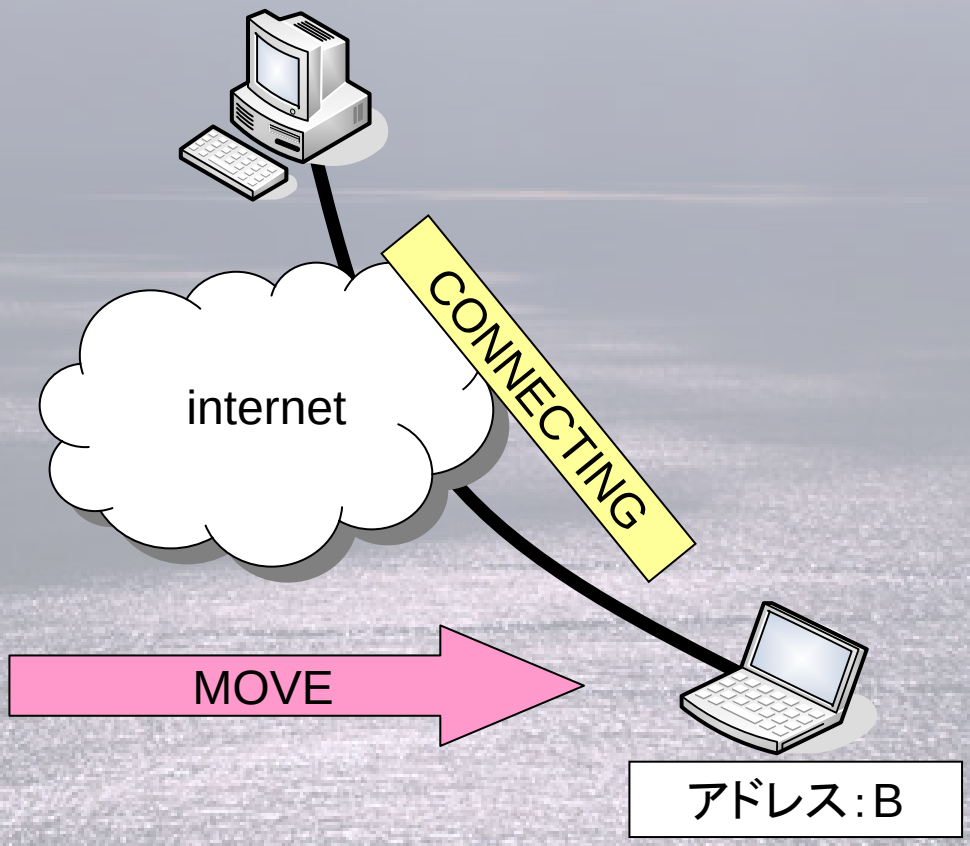




アドレス：B

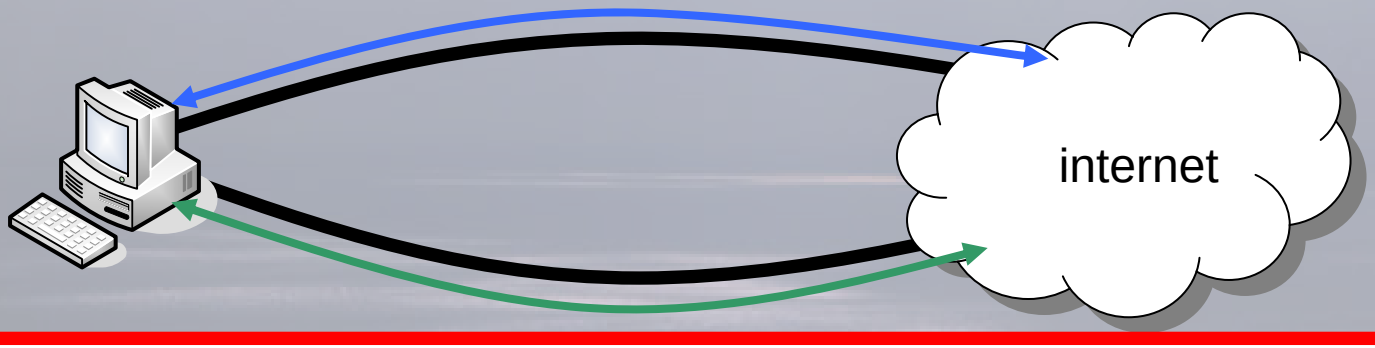
通信を継続したまま移動することが可能

モビリティ

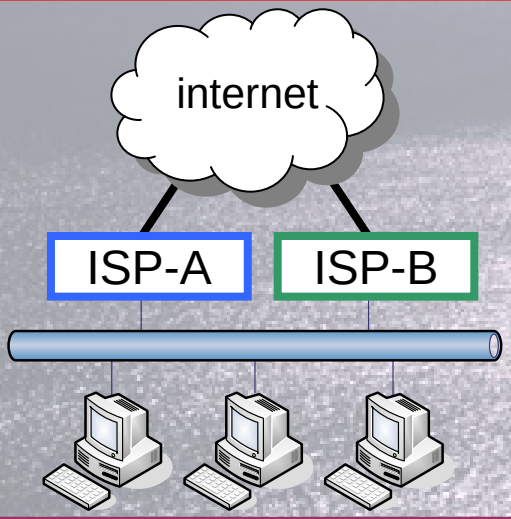


通信を継続したまま移動することが可能

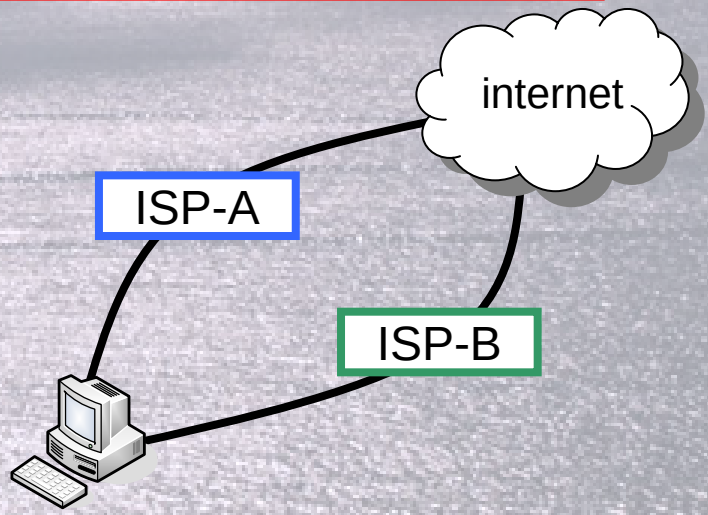
マルチホーミング



複数のインタフェースでネットワークと通信可能



サイトマルチホーム



エンドホストマルチホーム

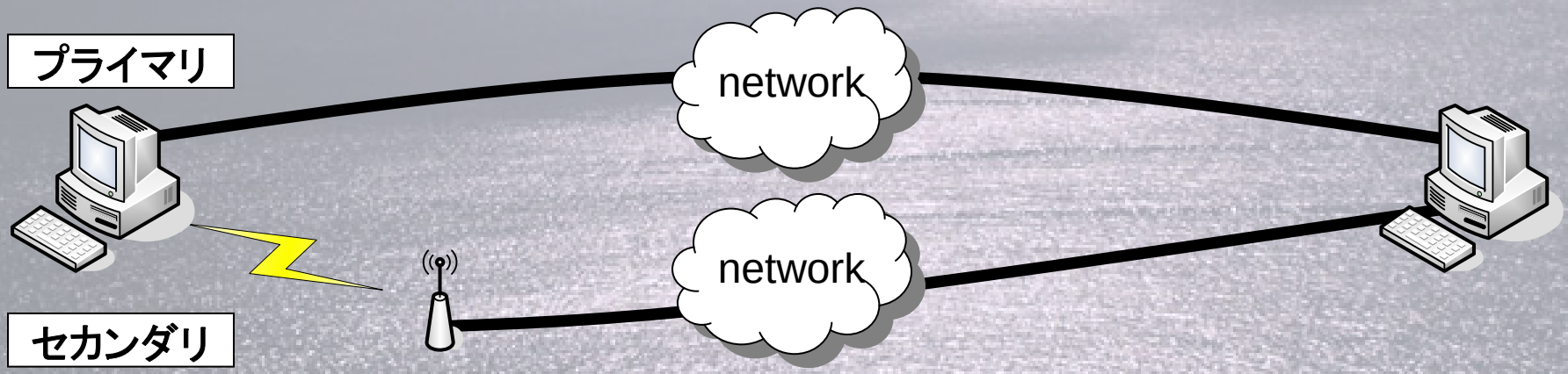
HIP以外の技術

- マルチホーミング
  - SCTP
- モビリティ
  - Mobile IPv6
  - LIN6

SCTP

マルチホーミングを実現

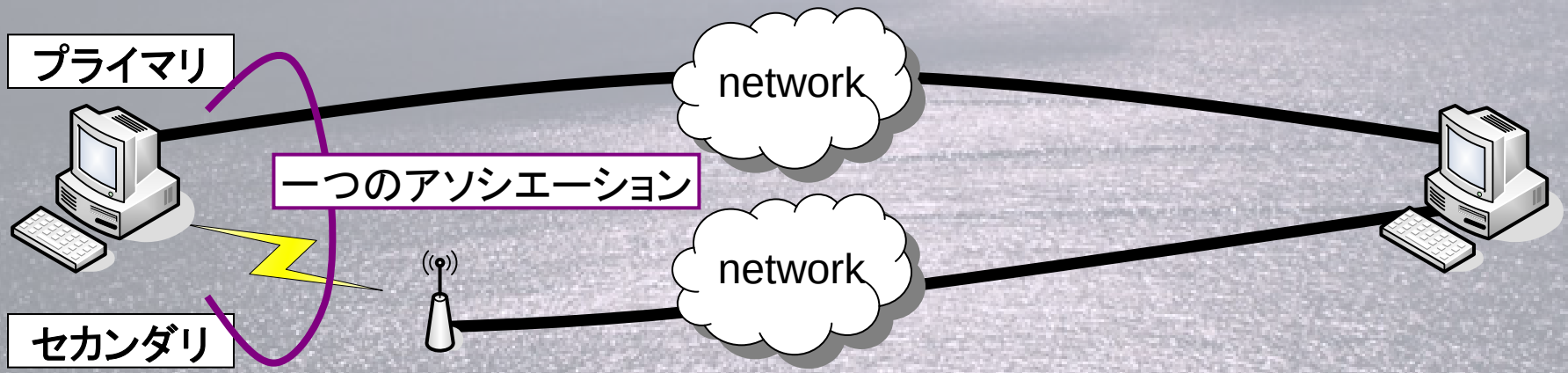
新しいトランスポートレイヤプロトコル



SCTP

マルチホーミングを実現

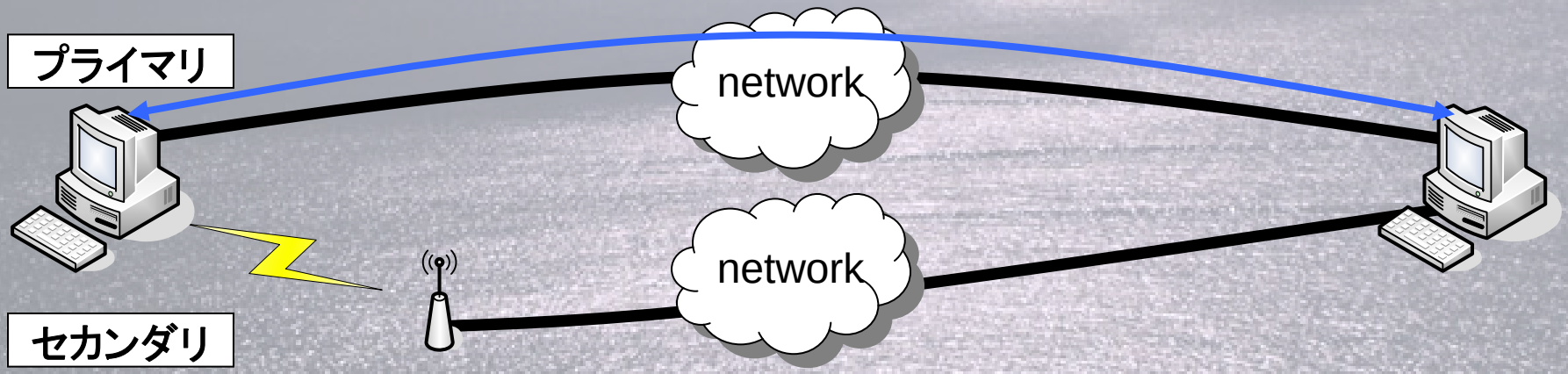
新しいトランスポートレイヤプロトコル



SCTP

マルチホーミングを実現

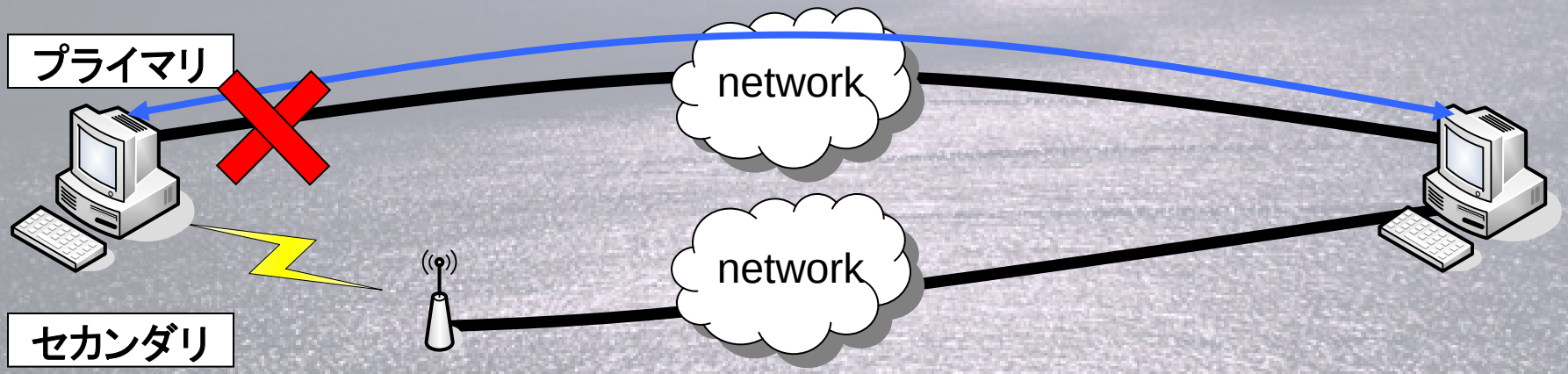
新しいトランスポートレイヤプロトコル



SCTP

マルチホーミングを実現

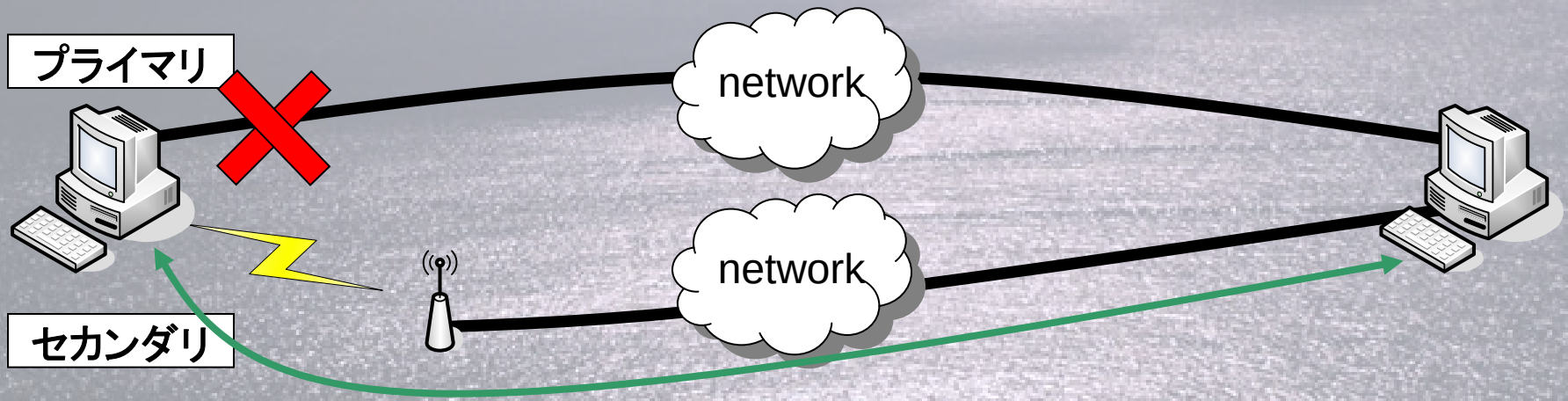
新しいトランスポートレイヤプロトコル



SCTP

マルチホーミングを実現

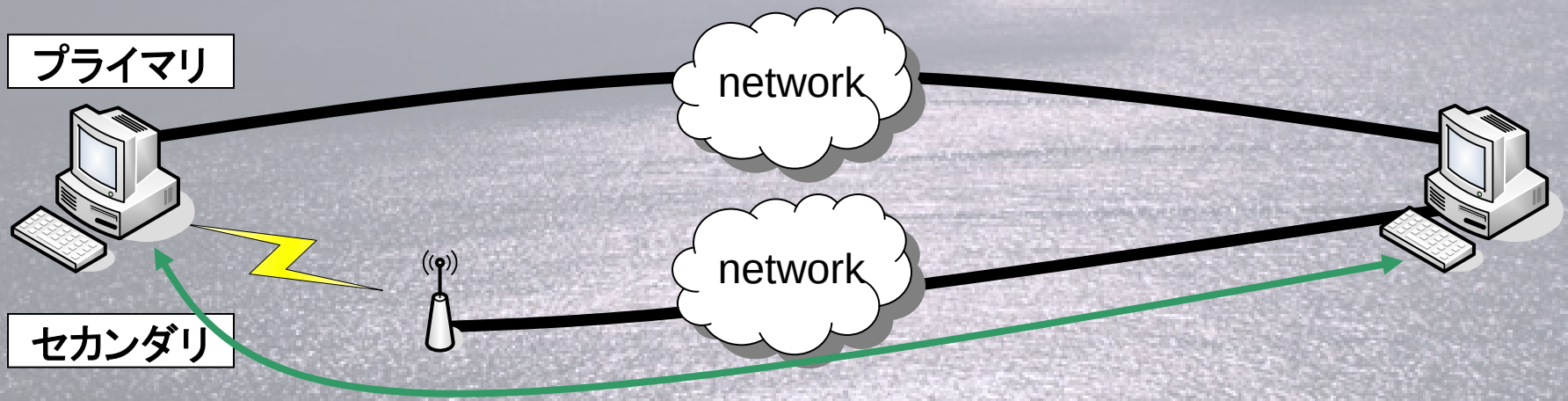
新しいトランスポートレイヤプロトコル



SCTP

マルチホーミングを実現

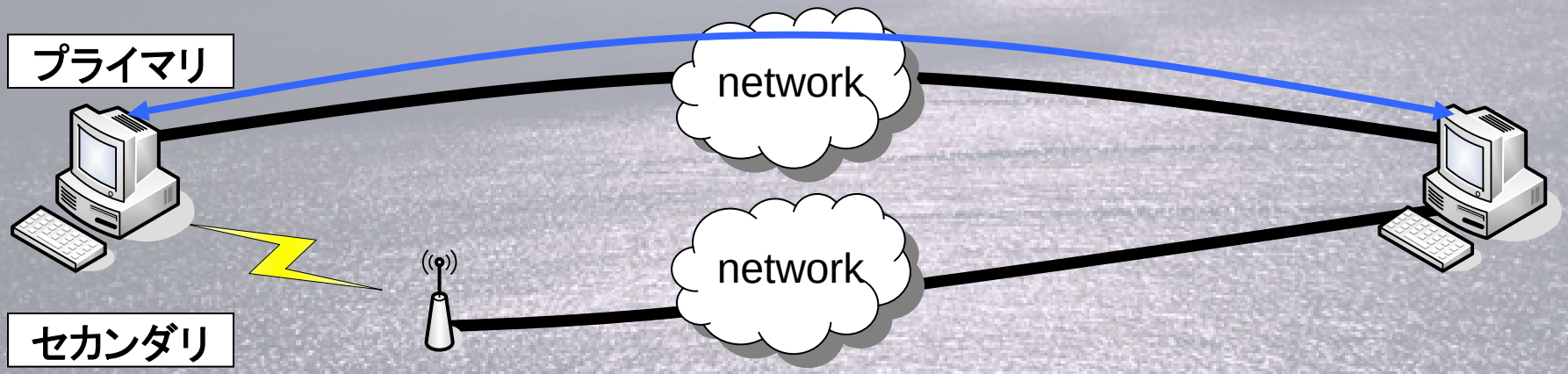
新しいトランスポートレイヤプロトコル



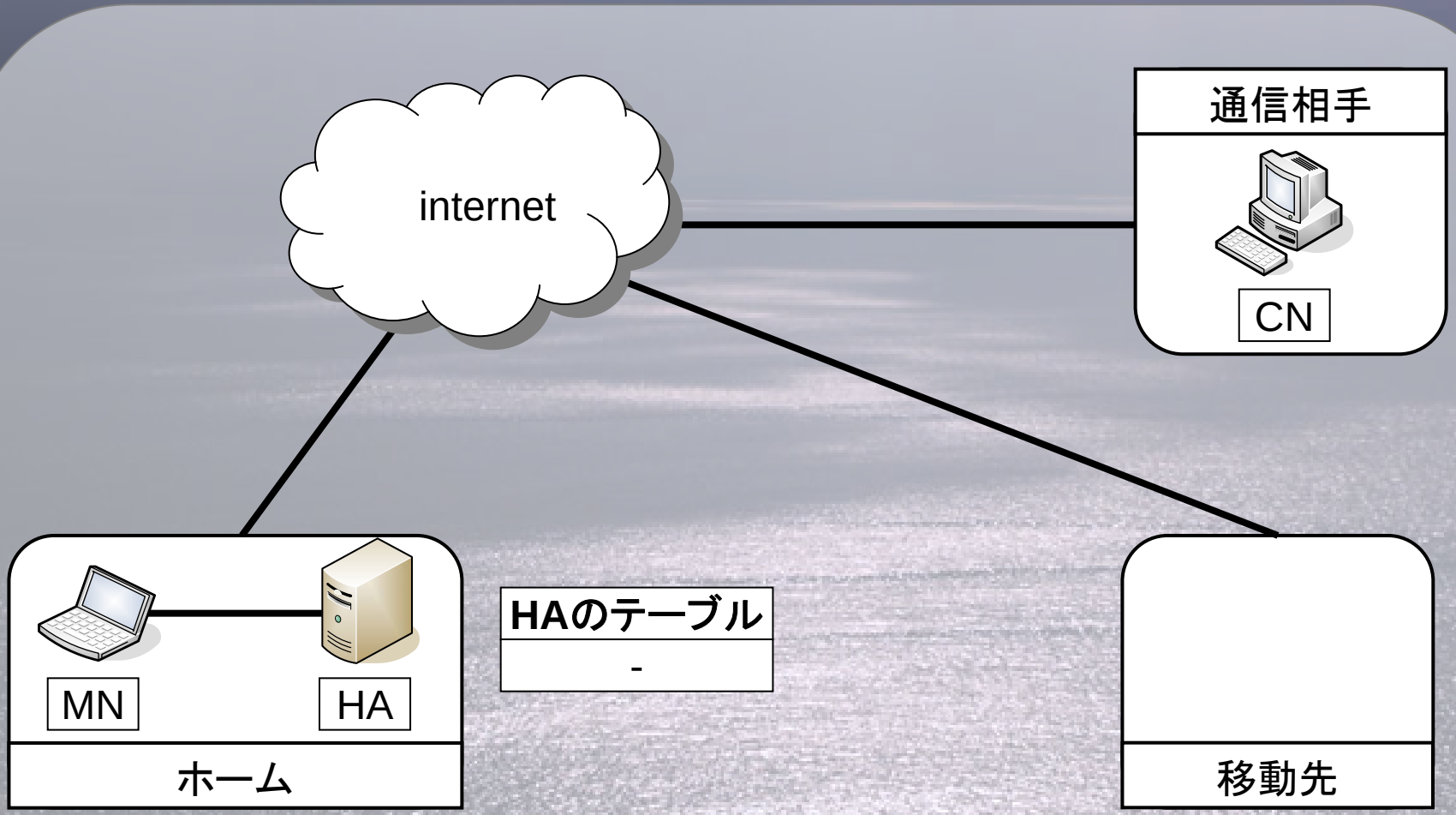
SCTP

マルチホーミングを実現

新しいトランスポートレイヤプロトコル

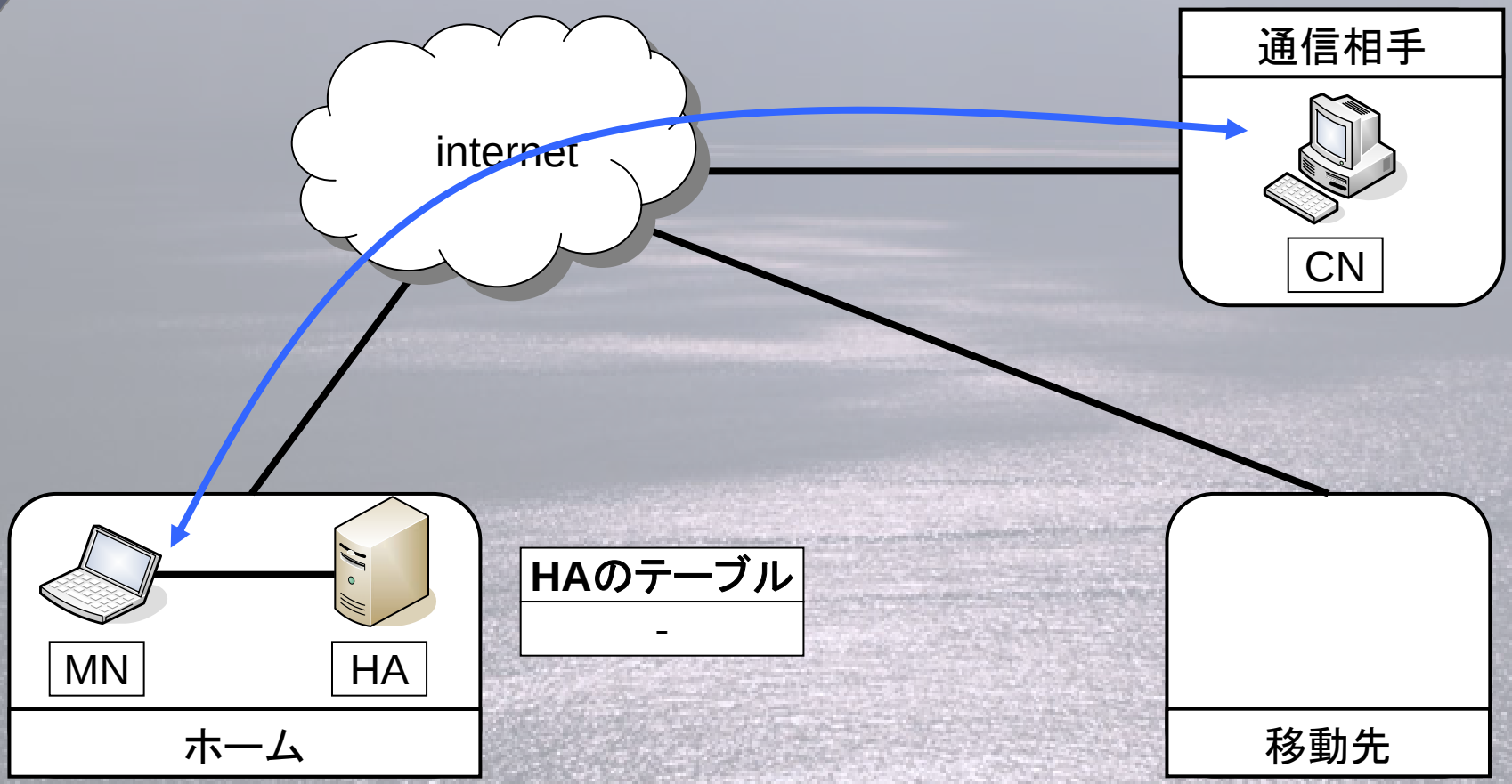


Mobile IP



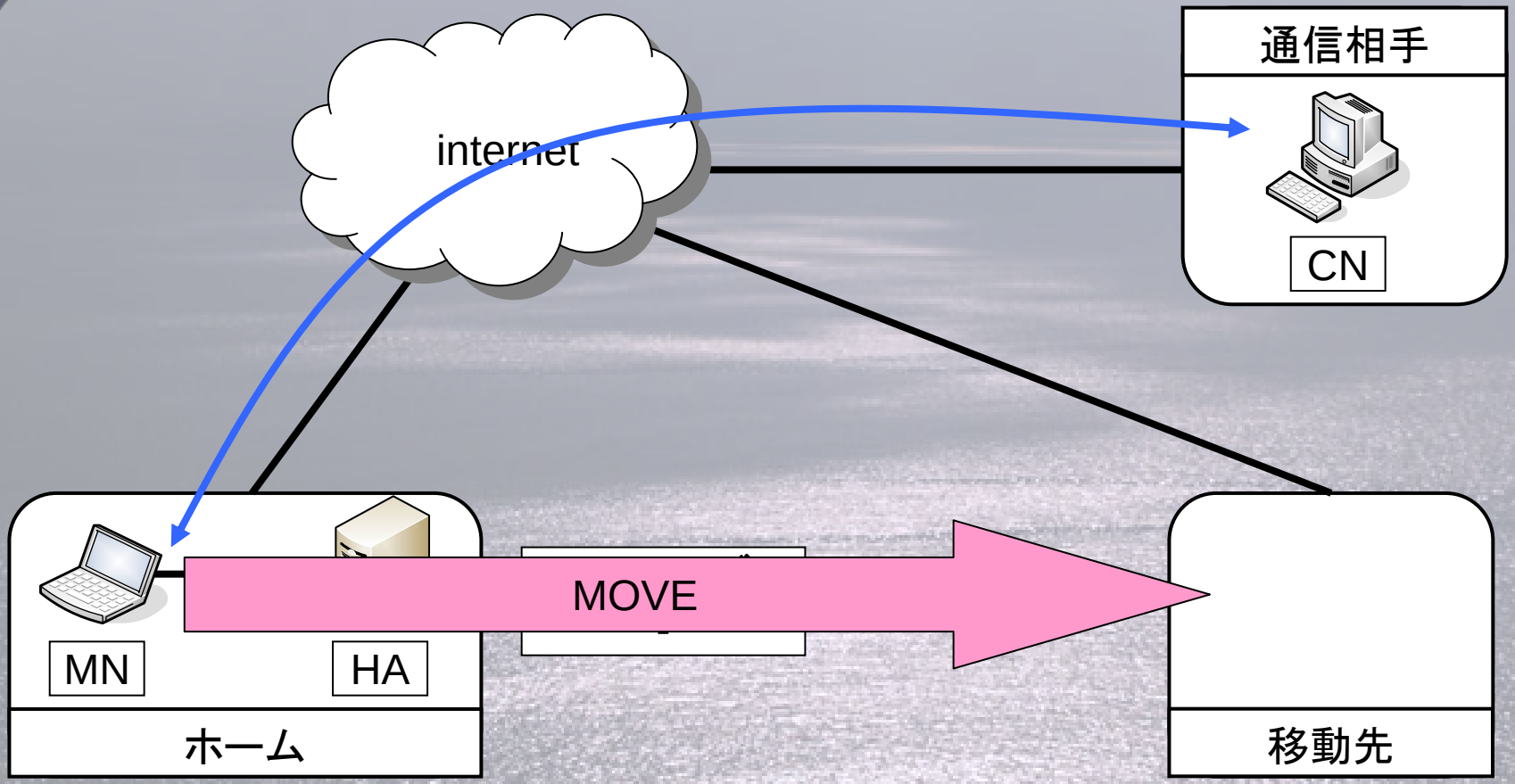
モビリティを実現

Mobile IP



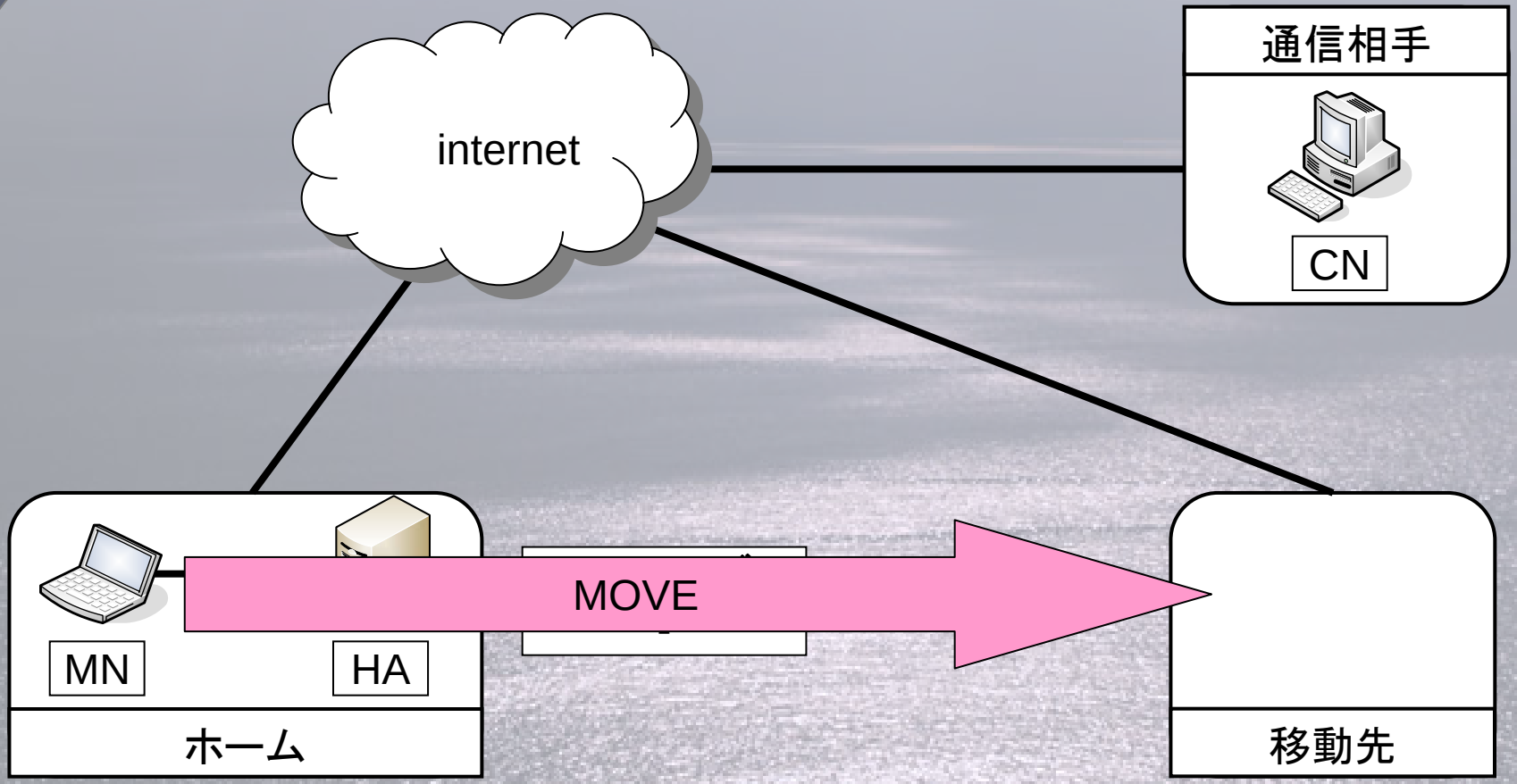
モビリティを実現

Mobile IP



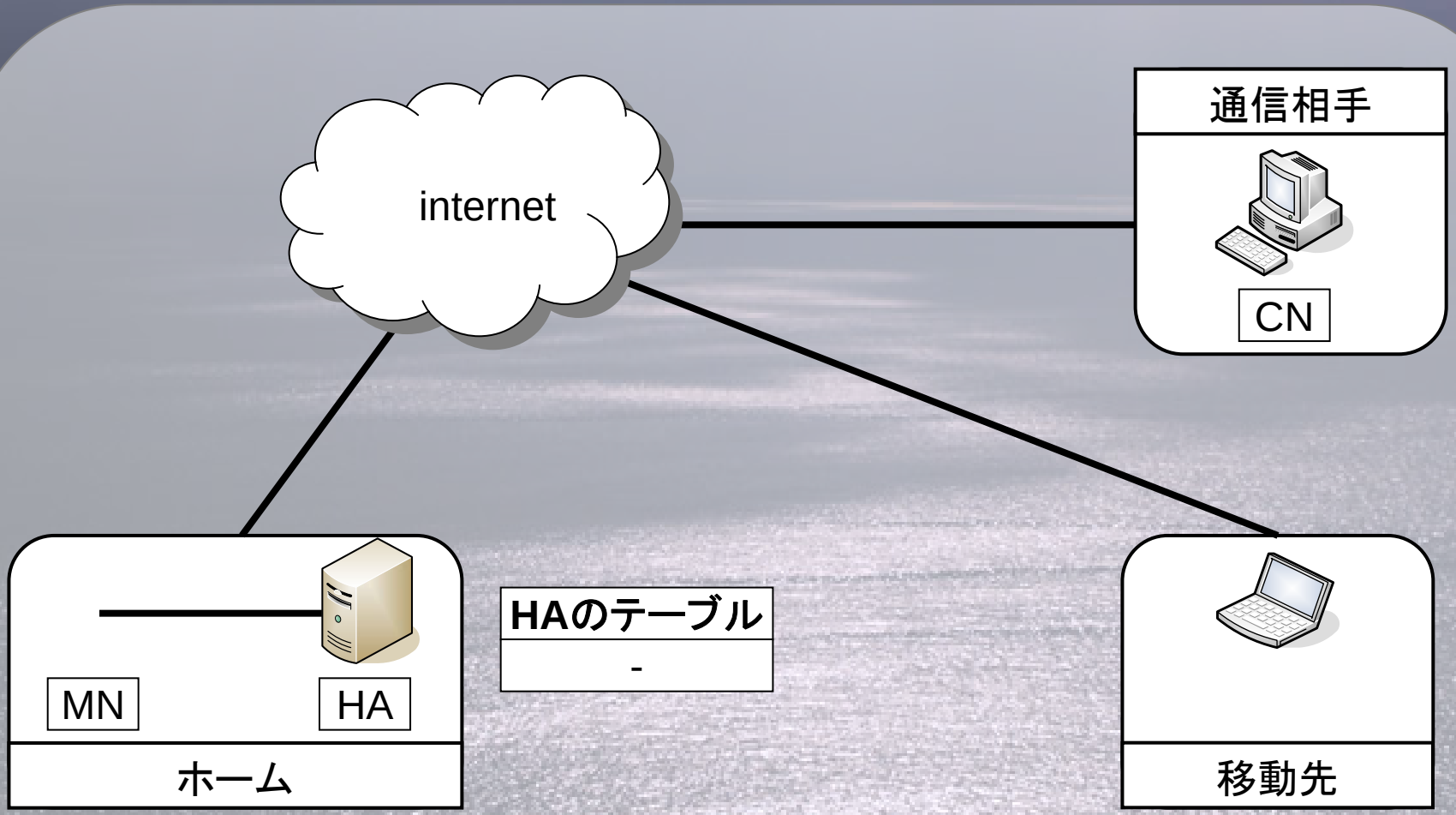
モビリティを実現

Mobile IP



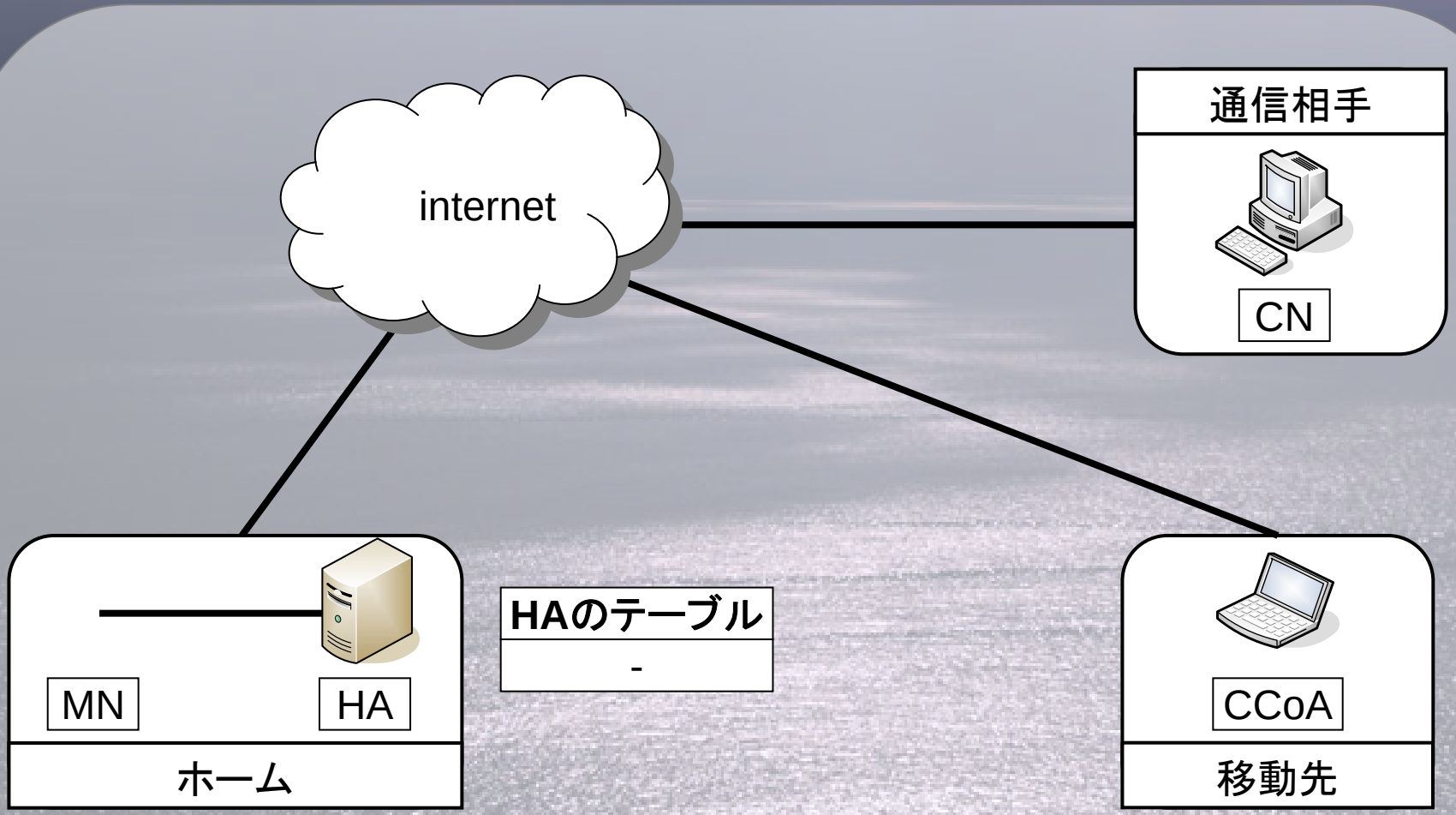
モビリティを実現

Mobile IP



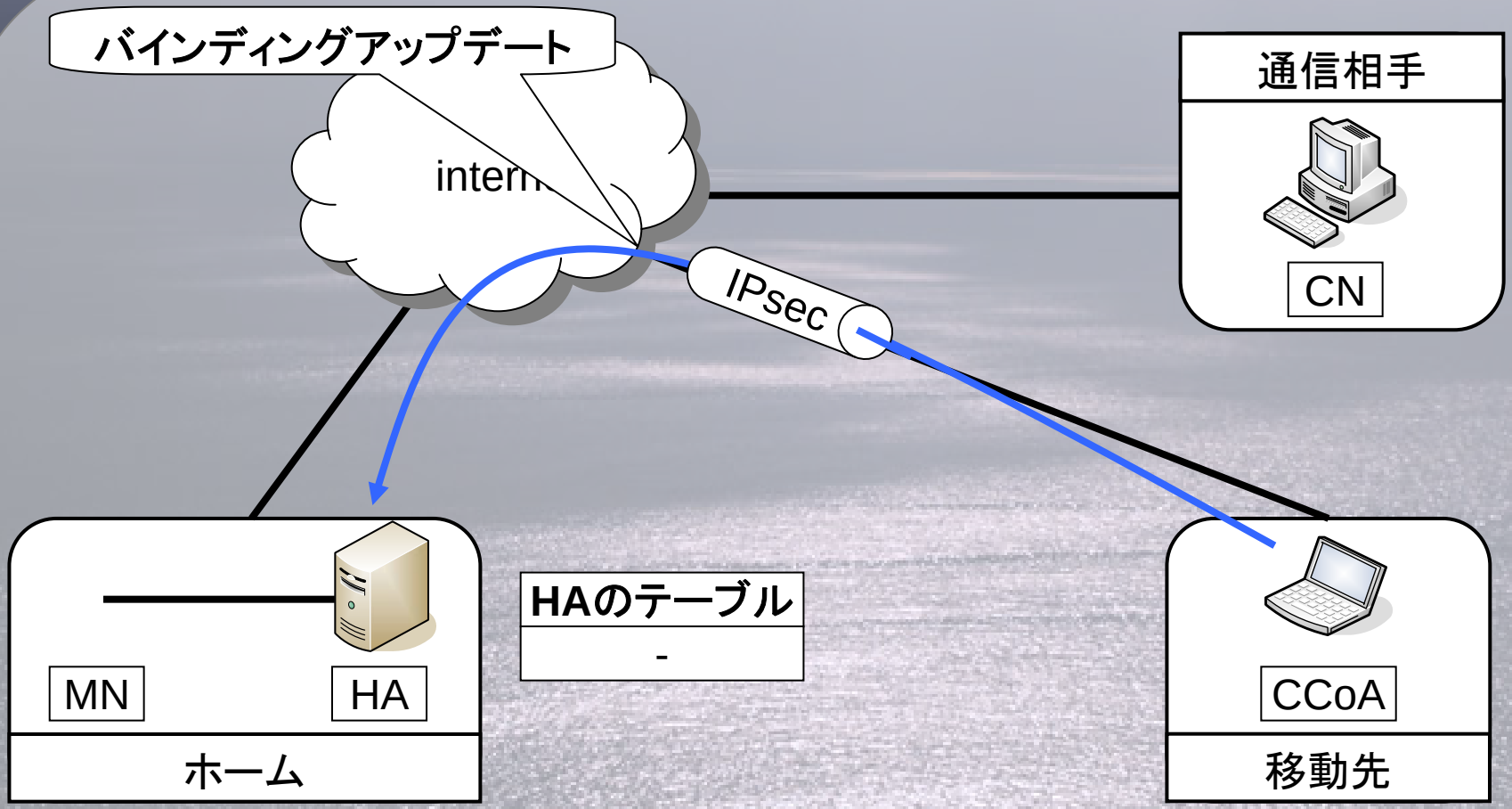
モビリティを実現

Mobile IP



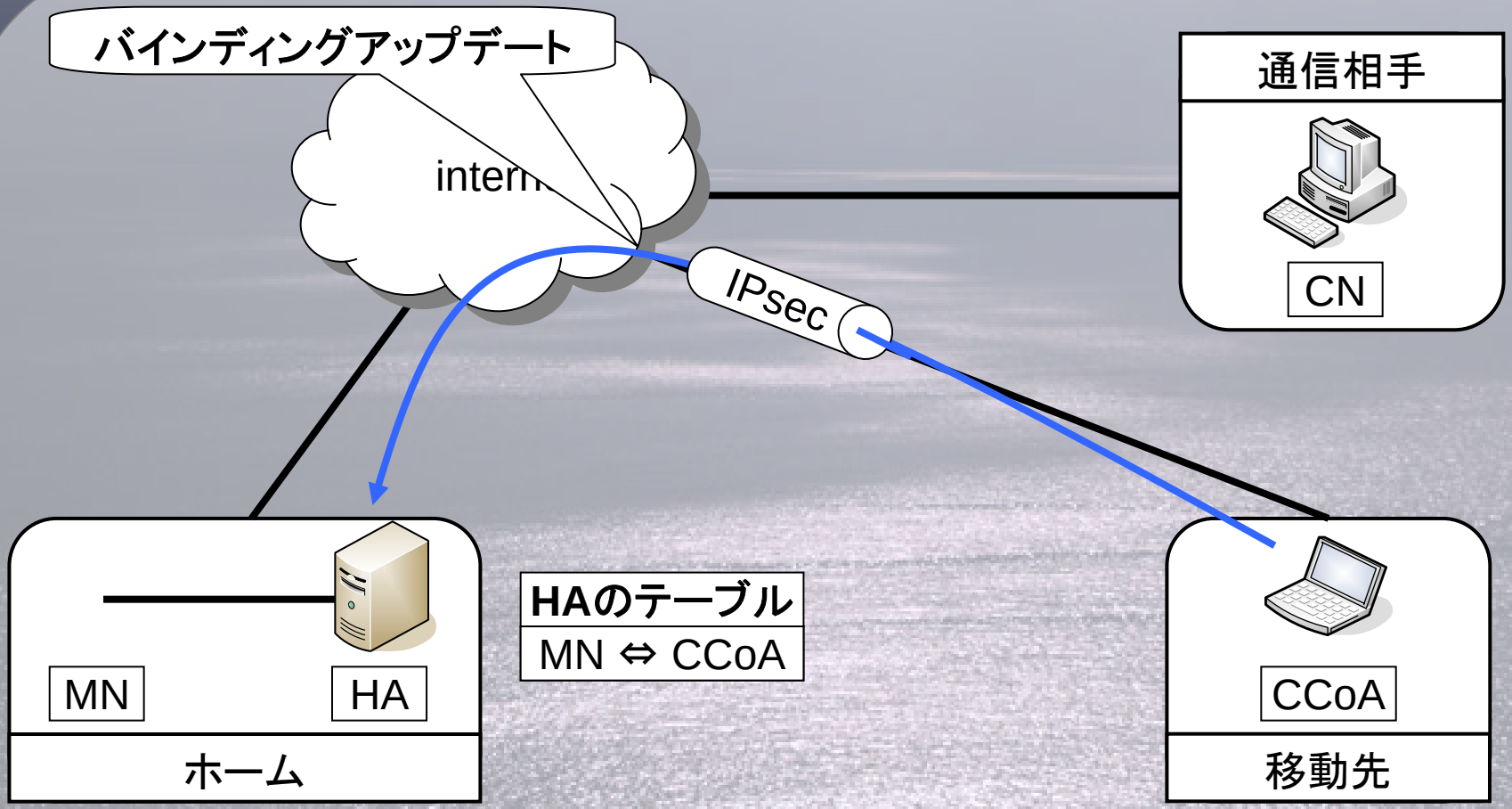
モビリティを実現

Mobile IP



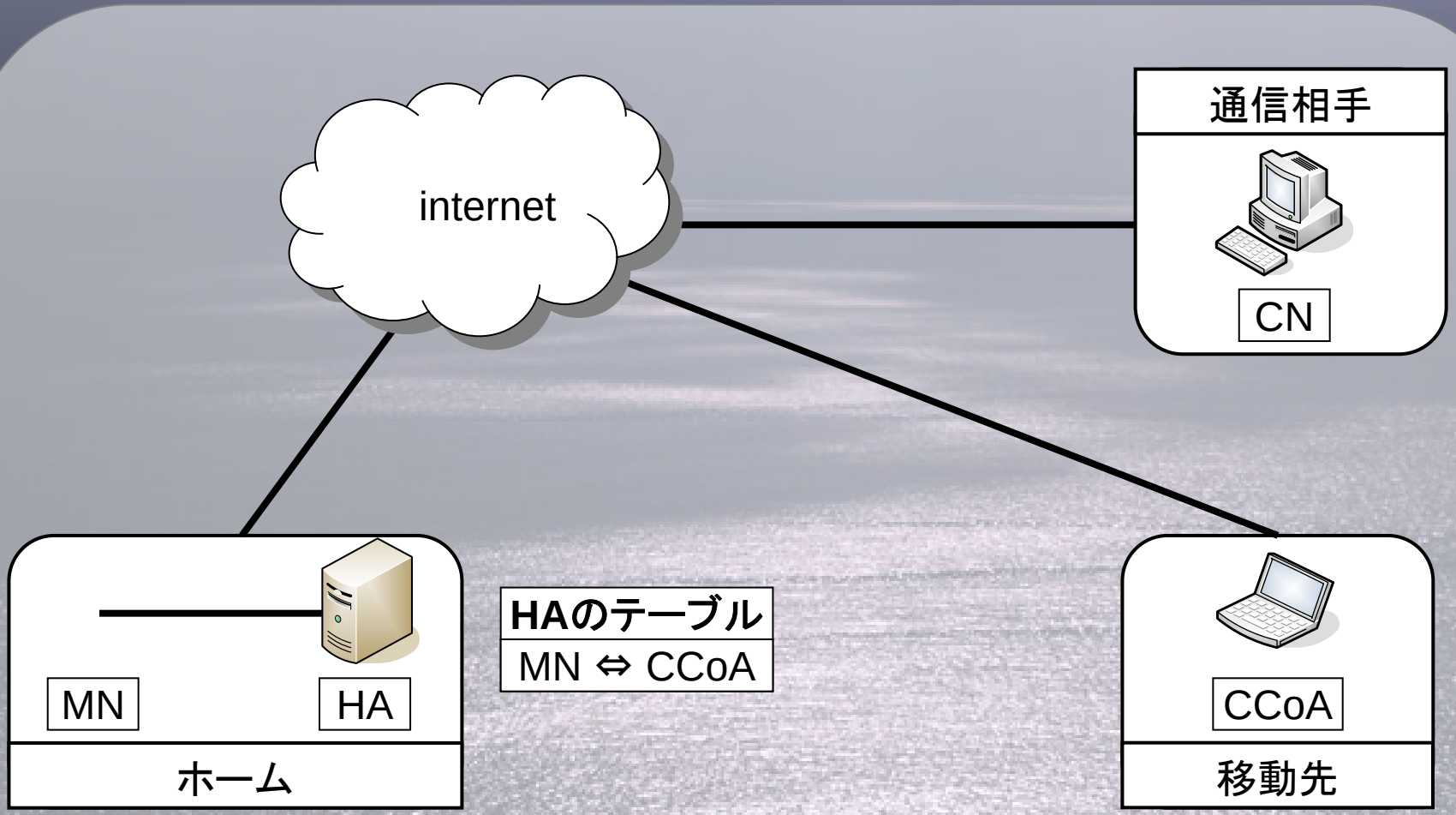
モビリティを実現

Mobile IP



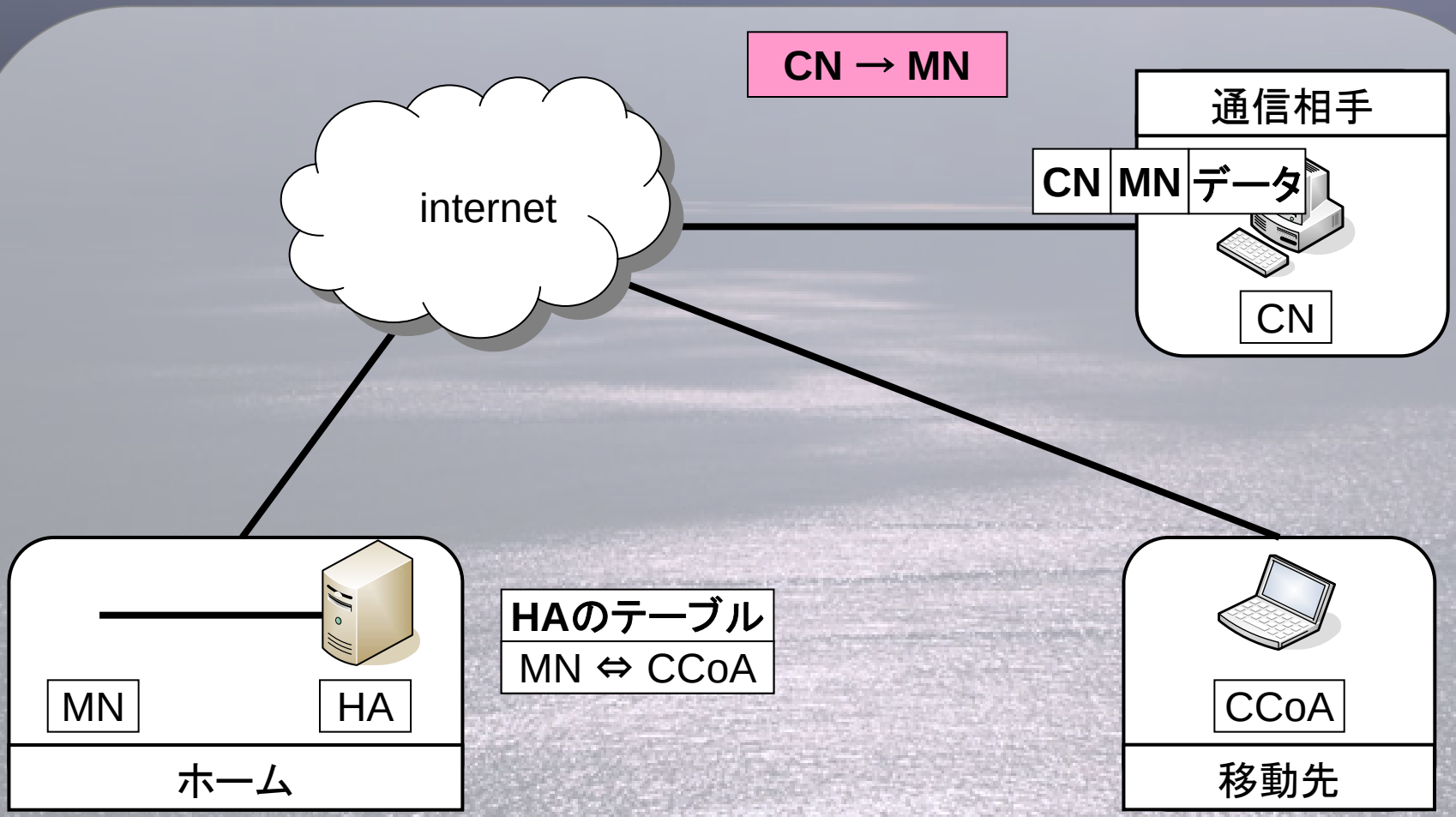
モビリティを実現

Mobile IP



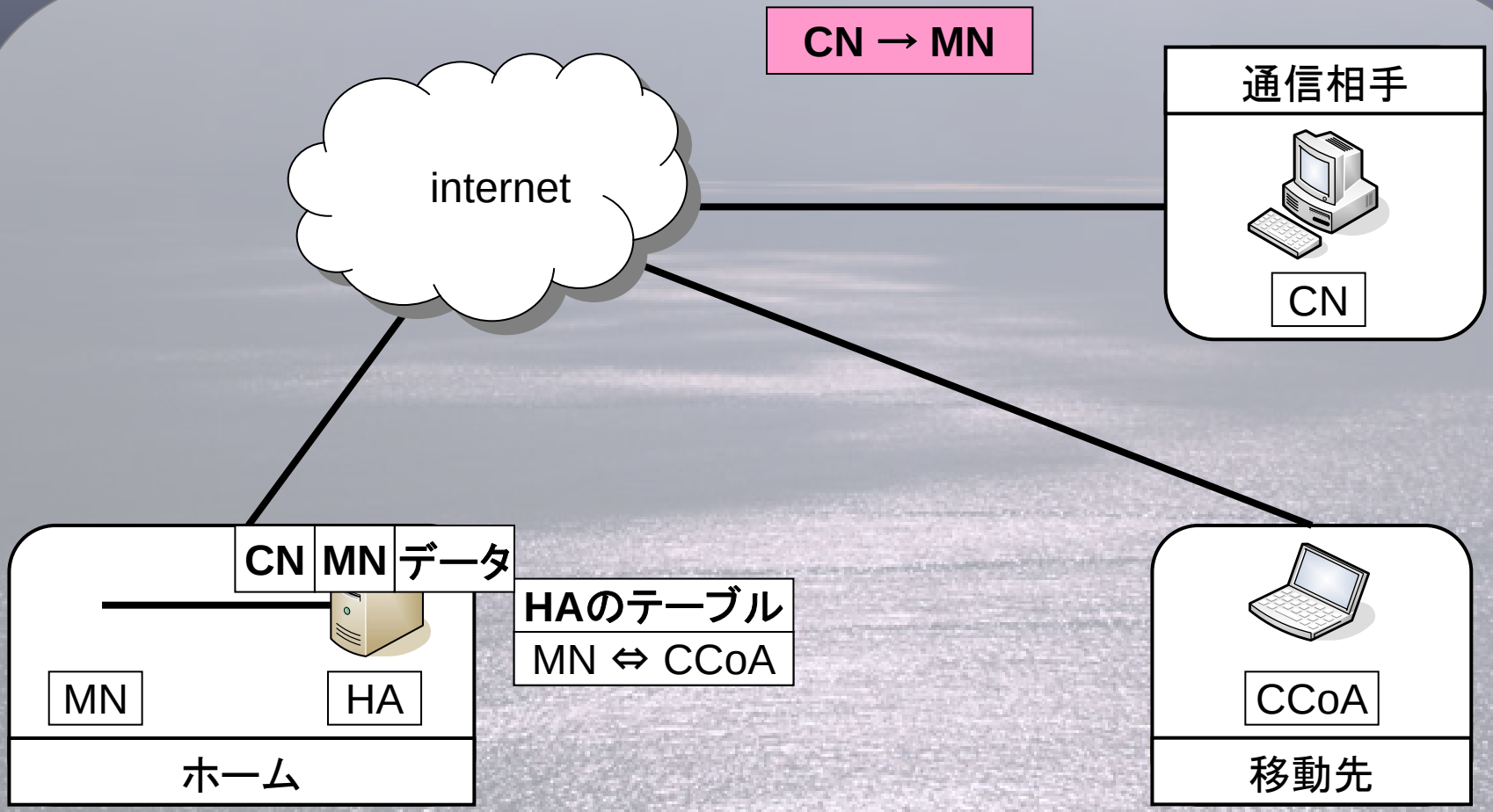
モビリティを実現

Mobile IP



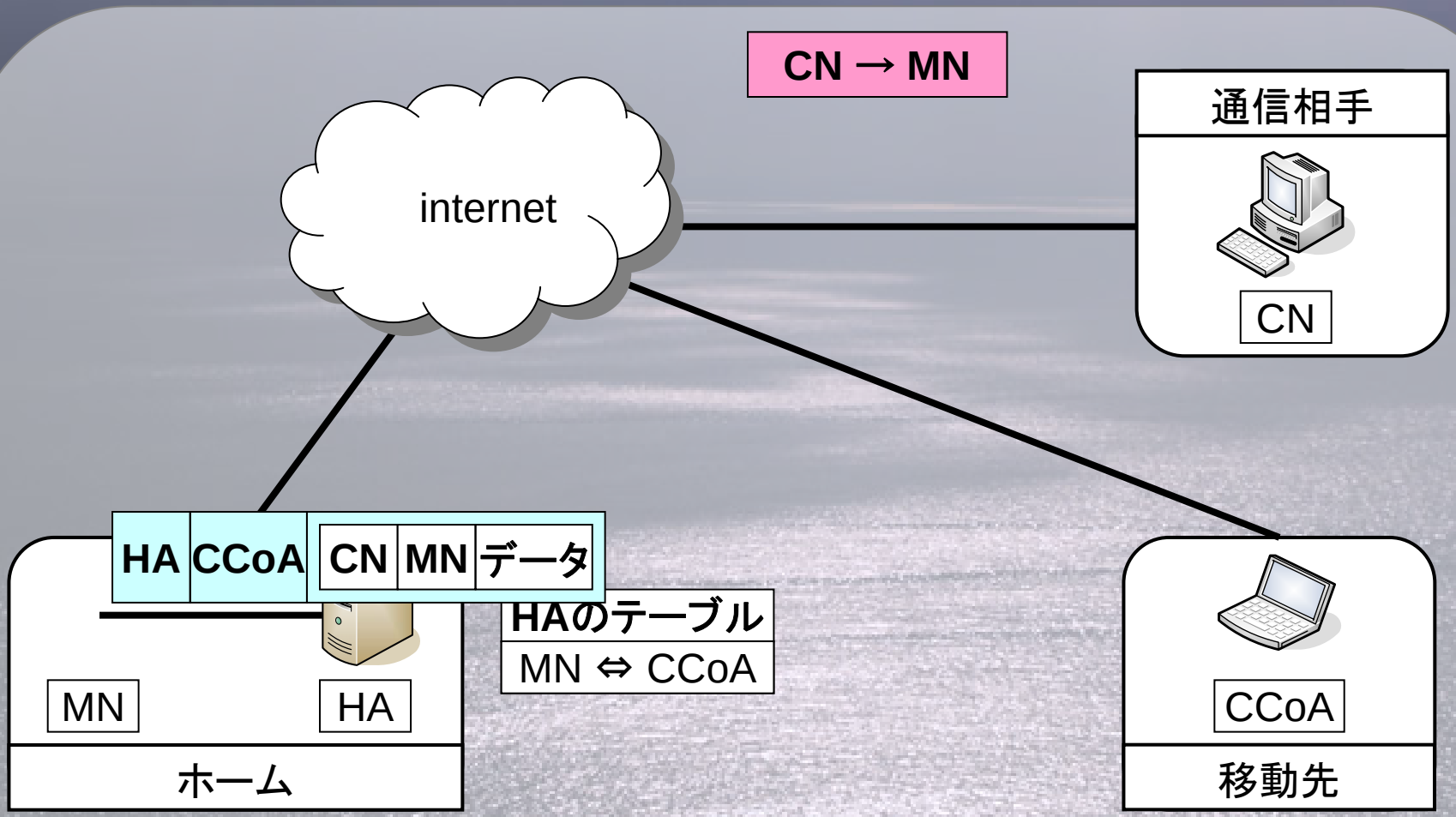
モビリティを実現

Mobile IP



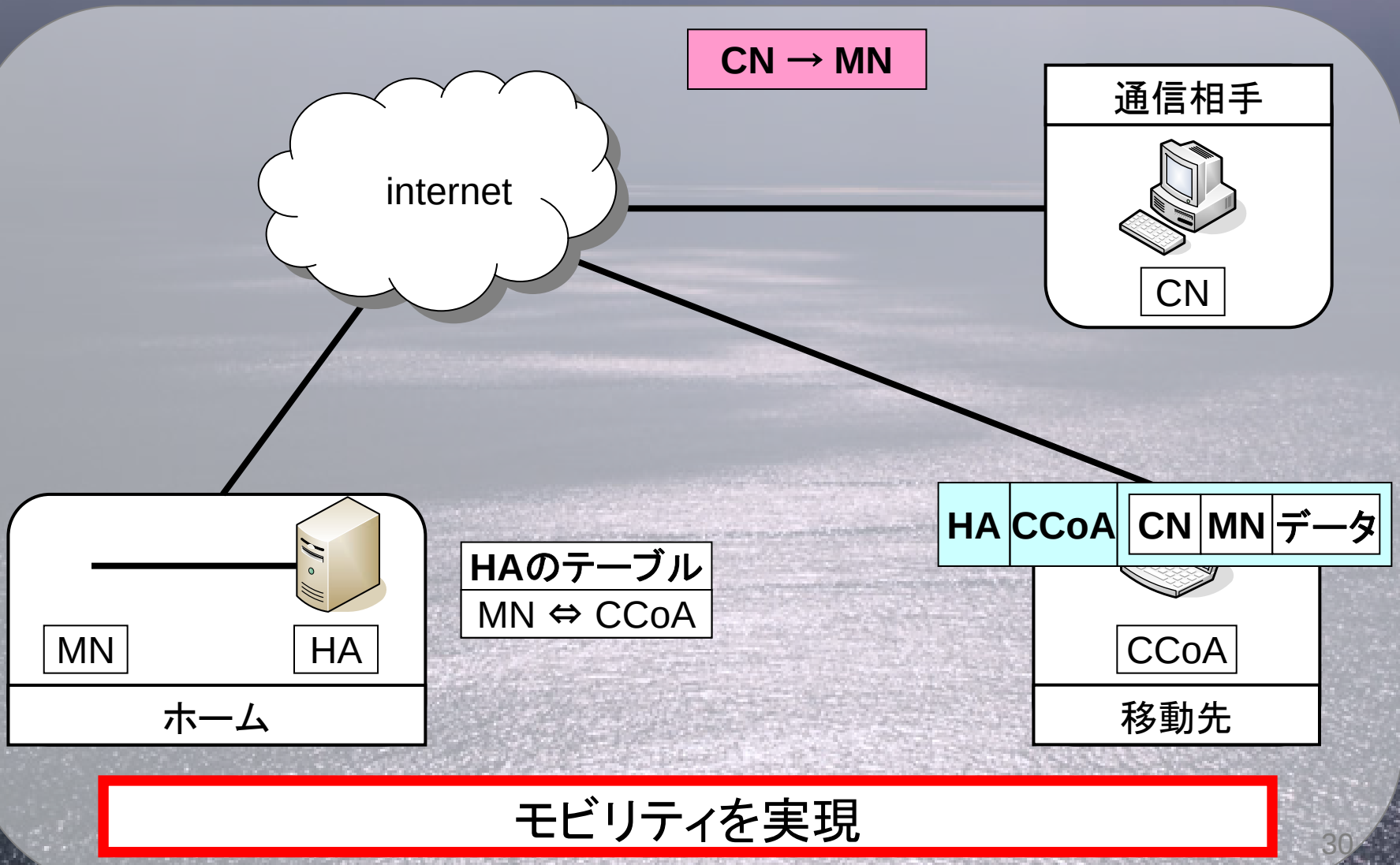
モビリティを実現

Mobile IP



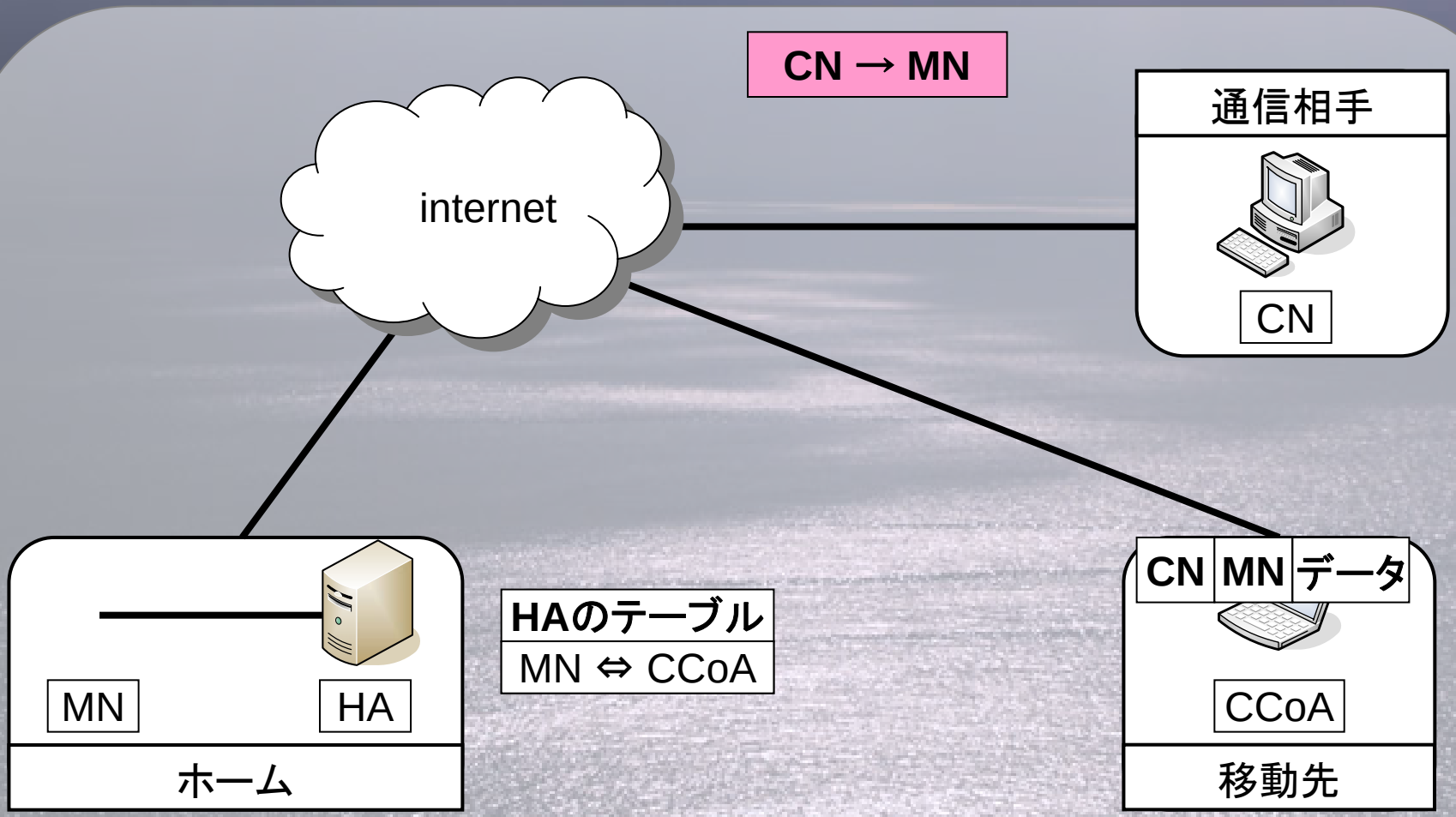
モビリティを実現

Mobile IP



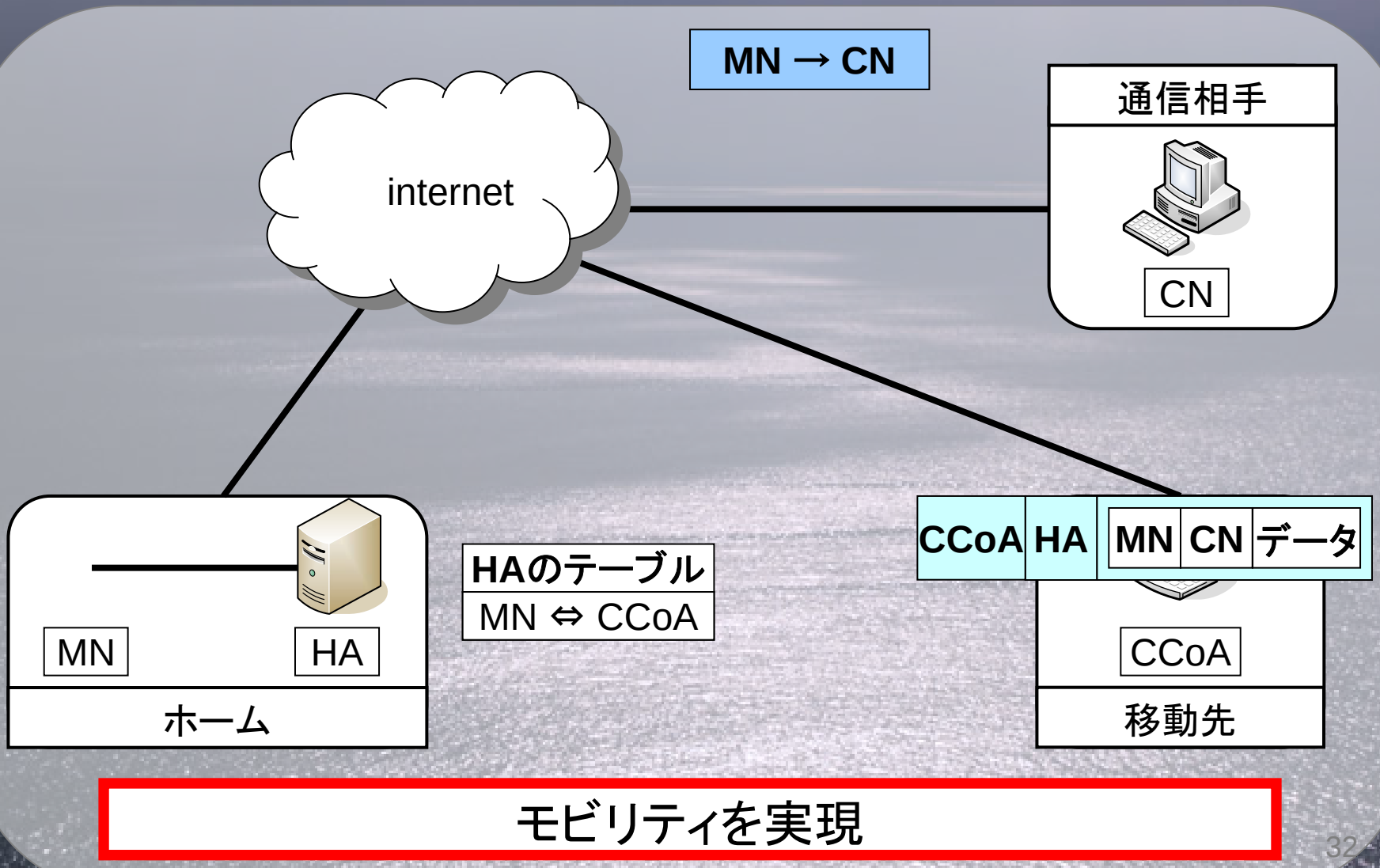
モビリティを実現

Mobile IP



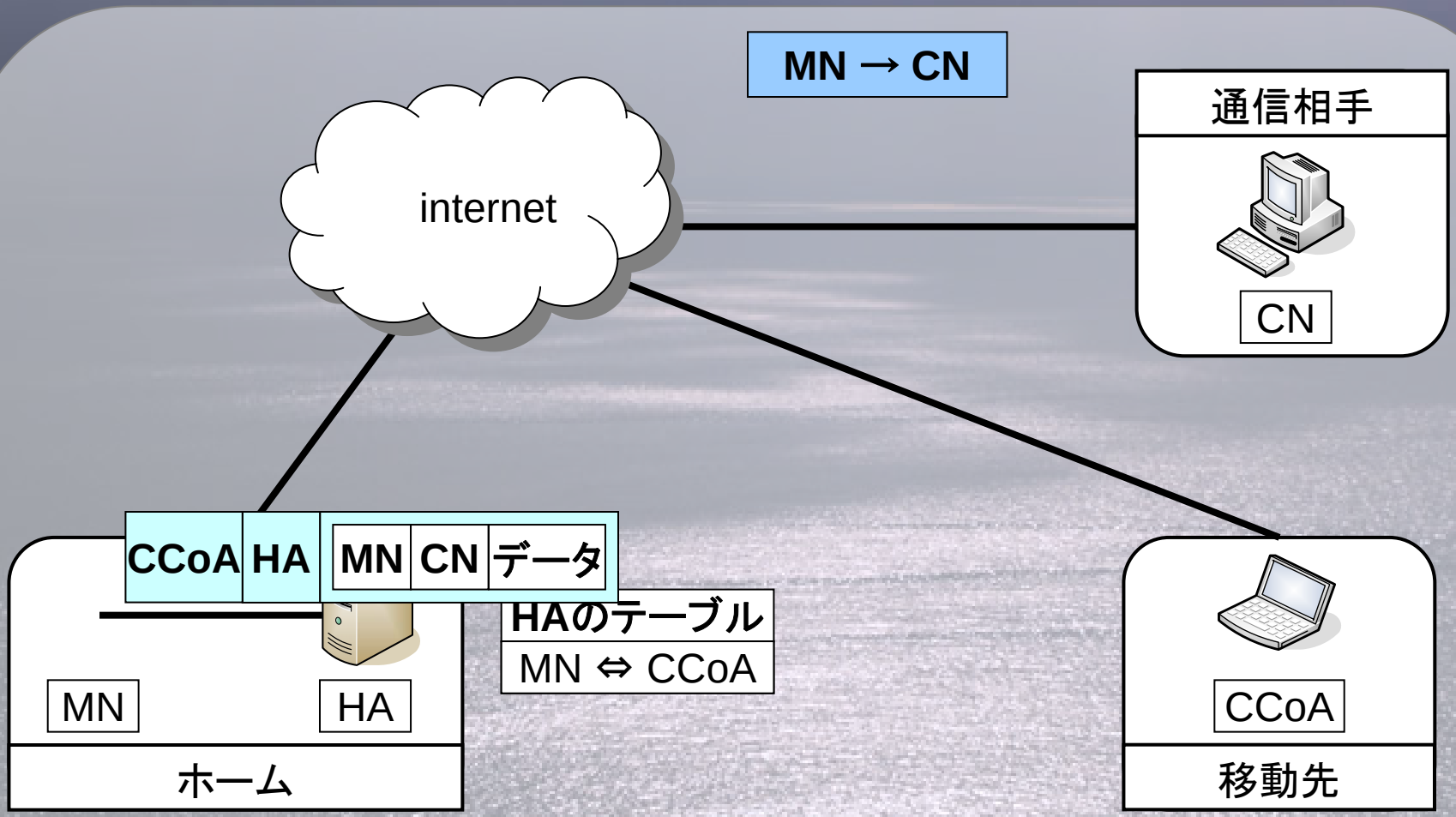
モビリティを実現

Mobile IP



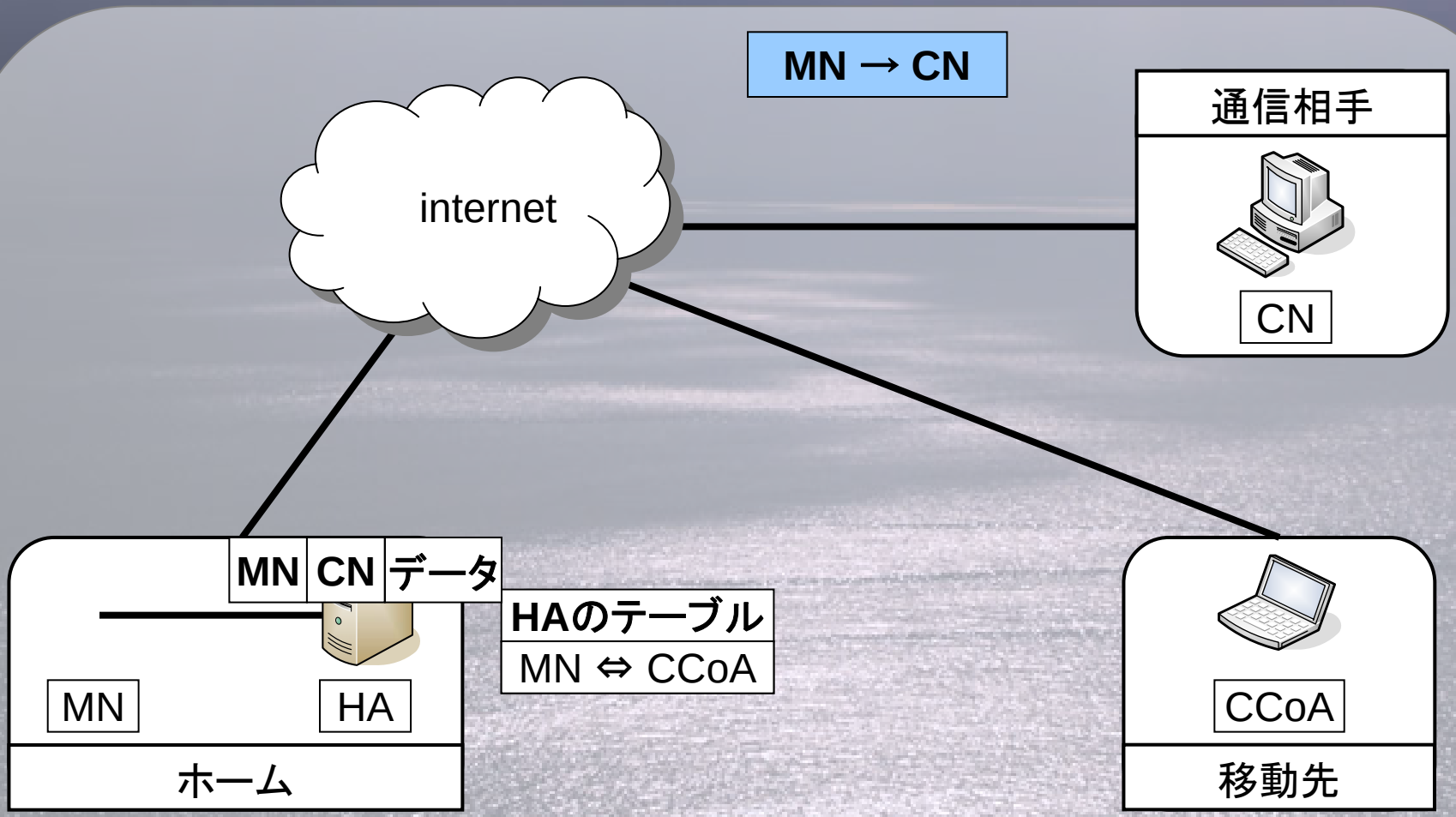
モビリティを実現

Mobile IP



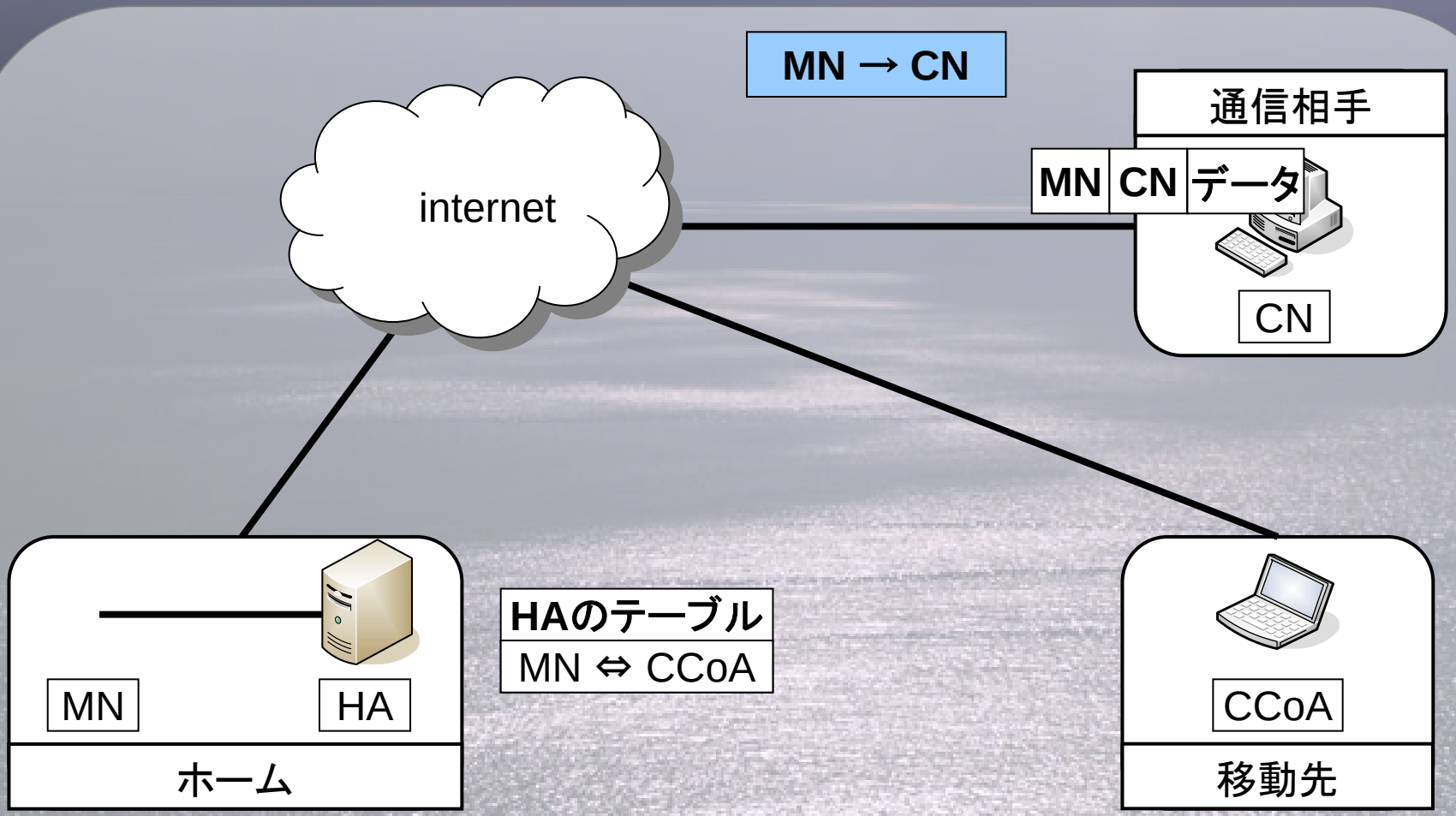
モビリティを実現

Mobile IP



モビリティを実現

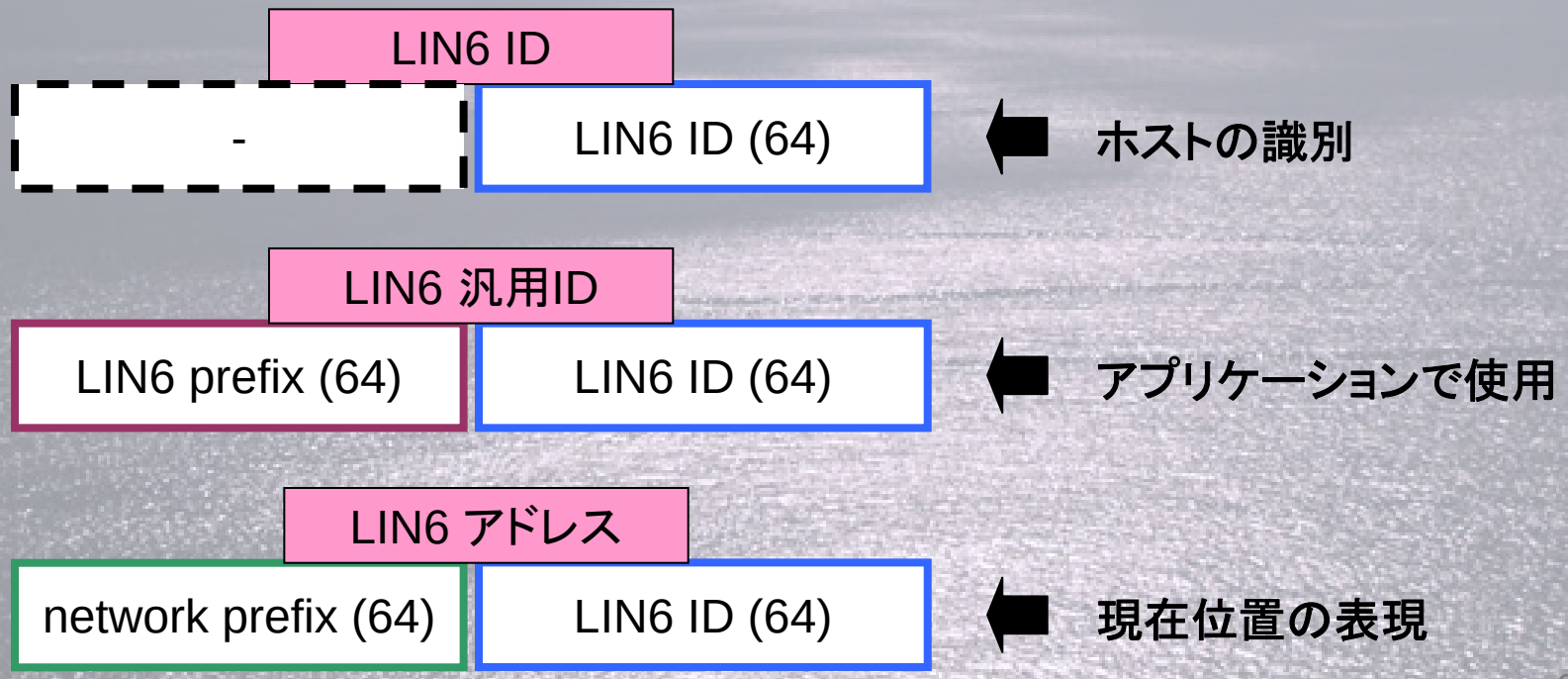
Mobile IP



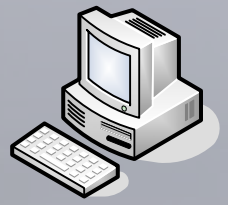
モビリティを実現

LIN6

- IPv6環境下で動作するプロトコル
- ホスト自身と現在位置を分ける



LIN6



Mapping Agent

ネットワークA

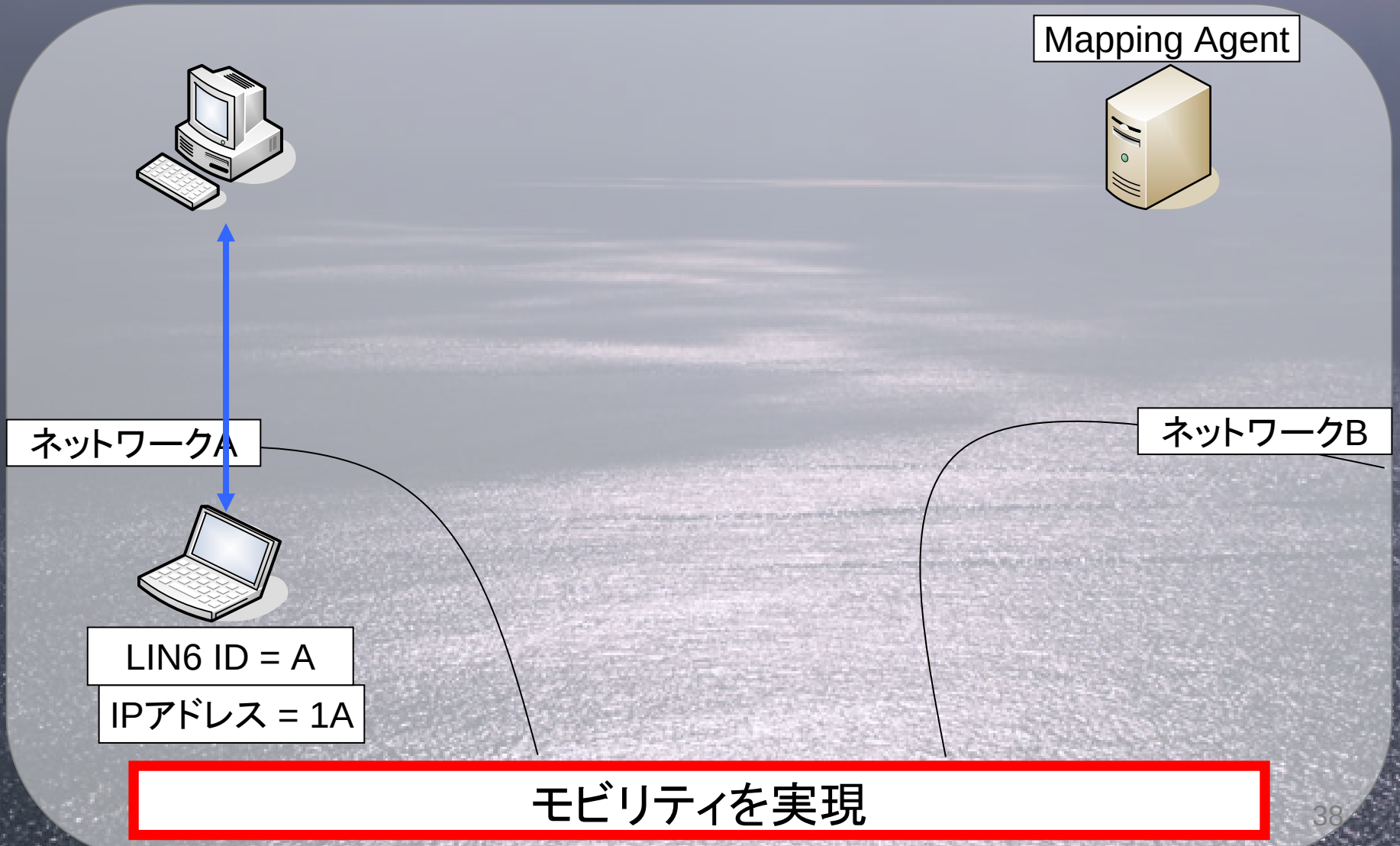
ネットワークB

LIN6 ID = A

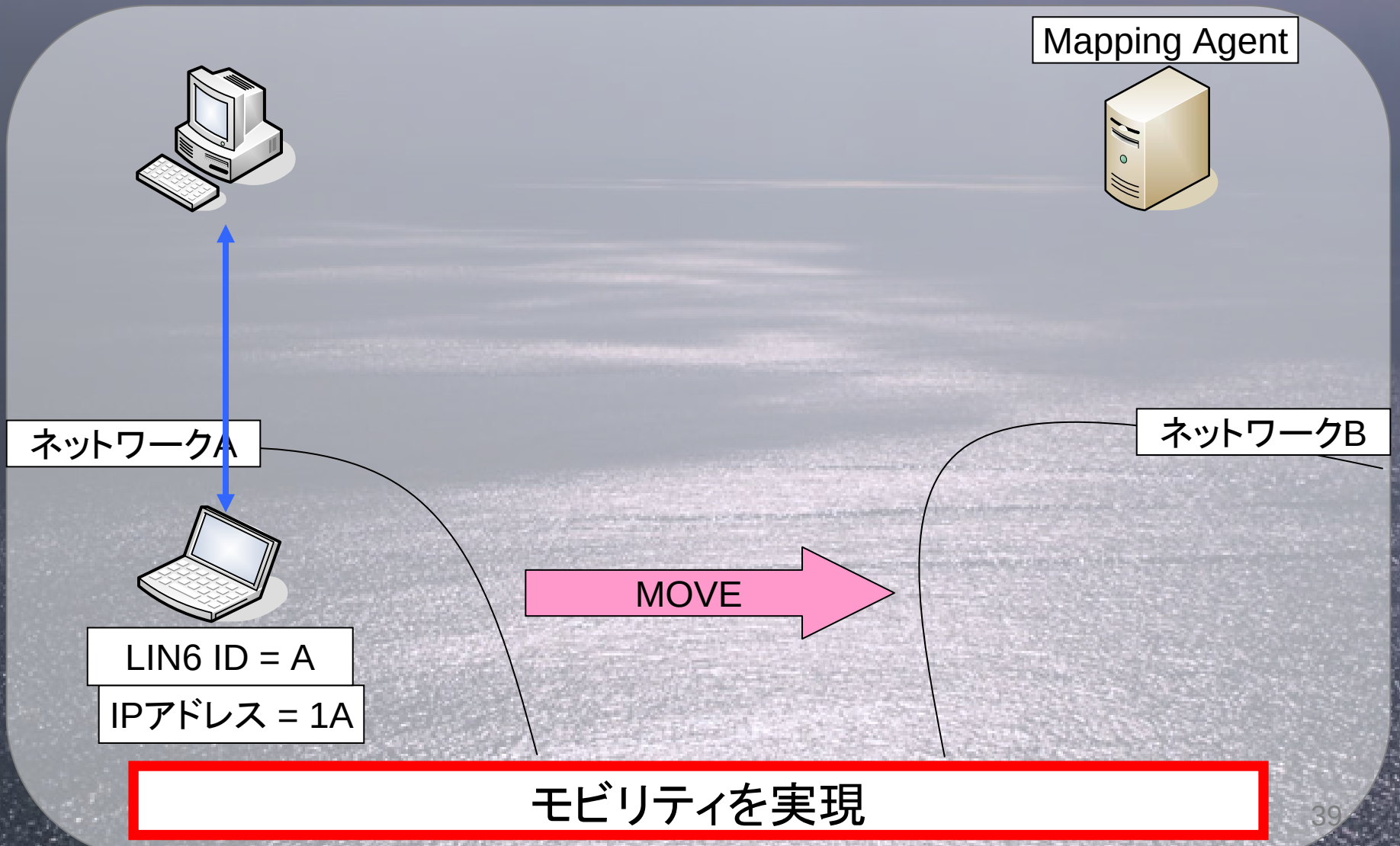
IPアドレス = 1A

モビリティを実現

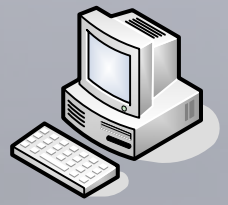
LIN6



LIN6


Mapping Agent
ネットワークA
ネットワークB
MOVE
LIN6 ID = A
IPアドレス = 1A
モビリティを実現

LIN6



Mapping Agent

ネットワークA

ネットワークB

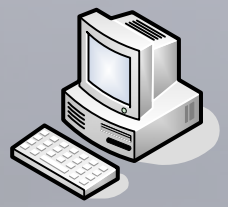
LIN6 ID = A

IPアドレス = 1A

MOVE

モビリティを実現

LIN6



Mapping Agent

ネットワークA

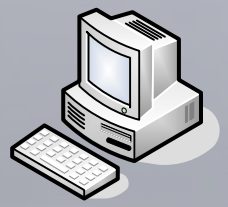
ネットワークB

MOVE

LIN6 ID = A

モビリティを実現

LIN6



Mapping Agent

ネットワークA

ネットワークB

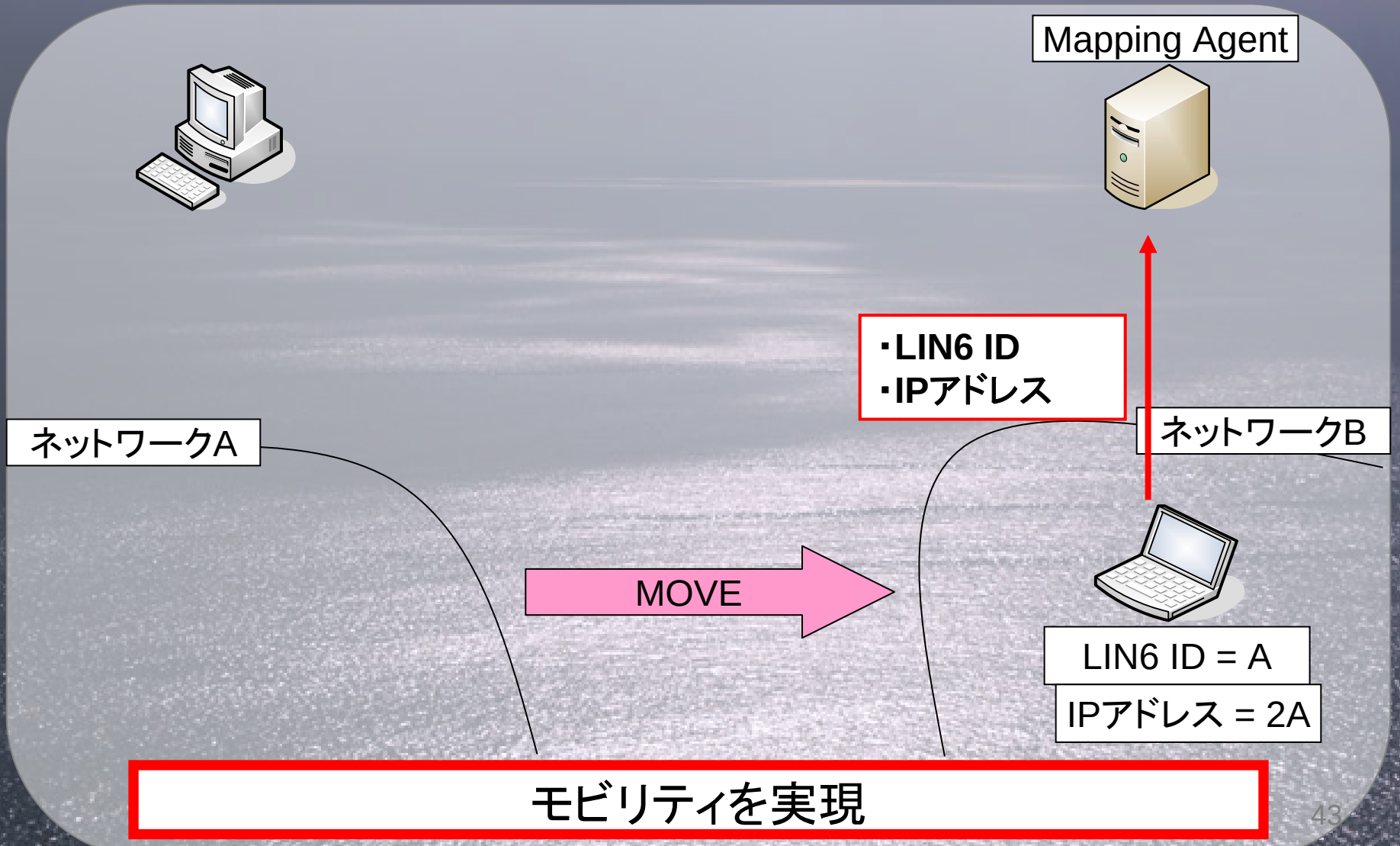
MOVE

LIN6 ID = A

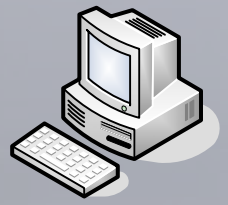
IPアドレス = 2A

モビリティを実現

LIN6


Mapping Agent
・LIN6 ID
・IPアドレス
ネットワークA
ネットワークB
MOVE
LIN6 ID = A
IPアドレス = 2A
モビリティを実現

LIN6



Mapping Agent

A ⇔ 2A

ネットワークA

ネットワークB

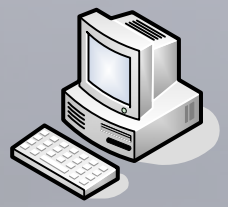
MOVE

LIN6 ID = A

IPアドレス = 2A

モビリティを実現

LIN6



今Aは何処に居ますか？

Mapping Agent

A ⇔ 2A

ネットワークA

ネットワークB

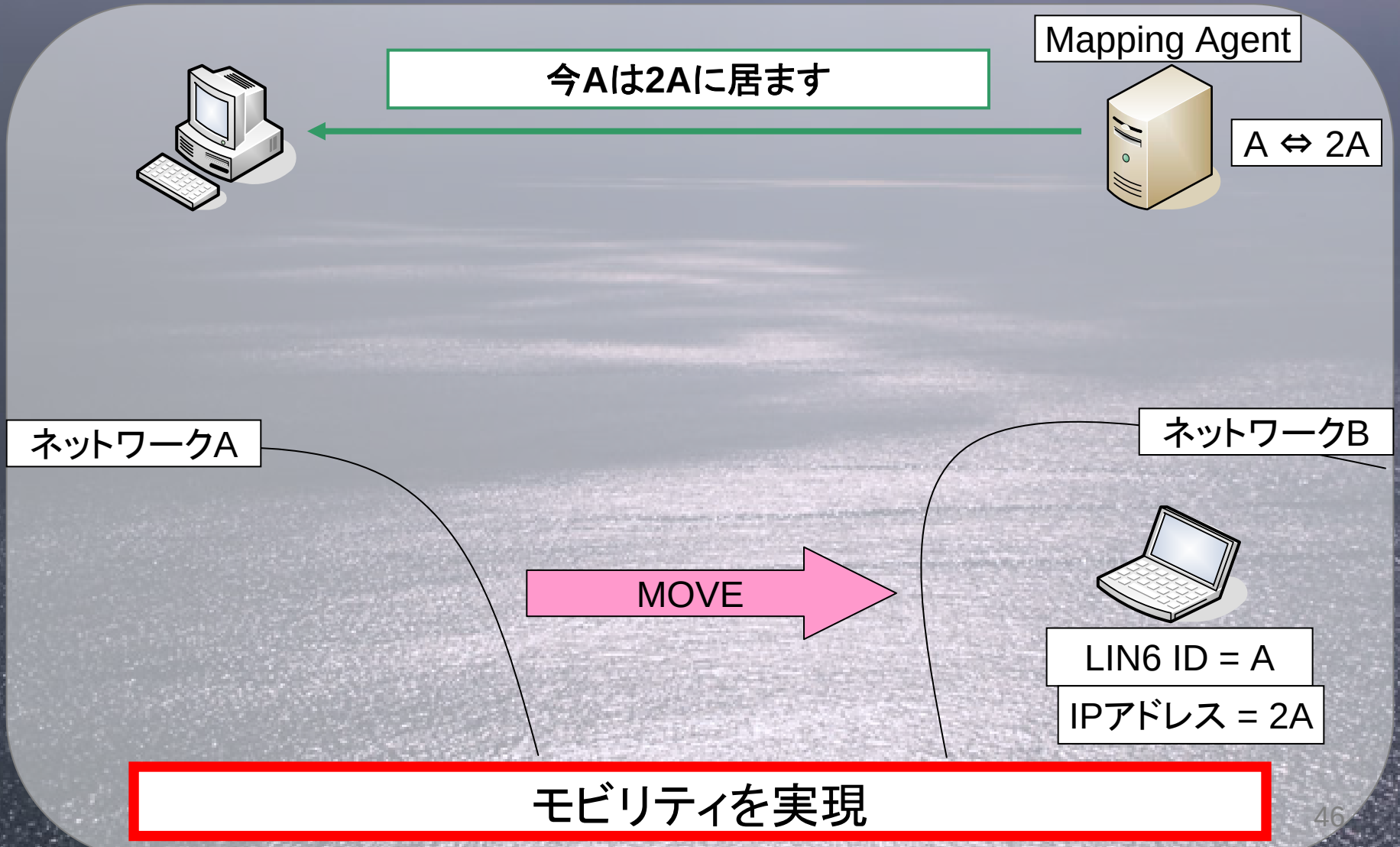
MOVE

LIN6 ID = A

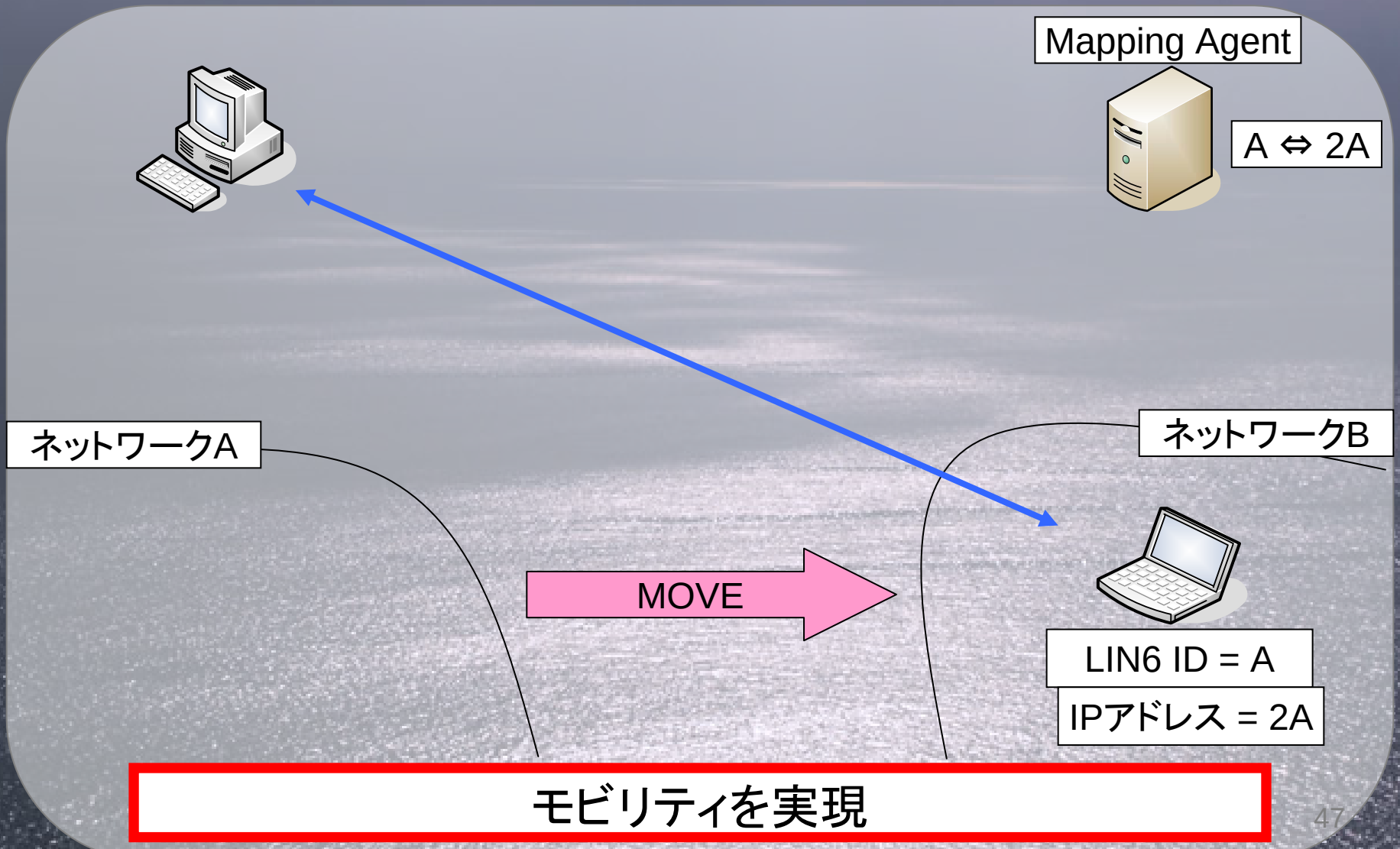
IPアドレス = 2A

モビリティを実現

LIN6


Mapping Agent
今Aは2Aに居ます
A ⇔ 2A
ネットワークA
ネットワークB
MOVE
LIN6 ID = A
IPアドレス = 2A
モビリティを実現

LIN6


Mapping Agent
A ⇔ 2A
ネットワークA
ネットワークB
MOVE
LIN6 ID = A
IPアドレス = 2A
モビリティを実現

HIP以外の技術の欠点

- SCTP
  - TCPが使えない
- Mobile IPv6
  - HAが必要で，経路が冗長
- LIN6
  - 第3者サービスのMAが必要

Host Identity Payload (HIP)

- ホストの識別と現在位置情報を分ける
  - ホストIDの導入
  - ホストIDに関するプロトコルの導入
  - ホストIDに関する処理を行うレイヤの導入
  - そのレイヤ同士のプロトコル(HLP)の導入

ホストID

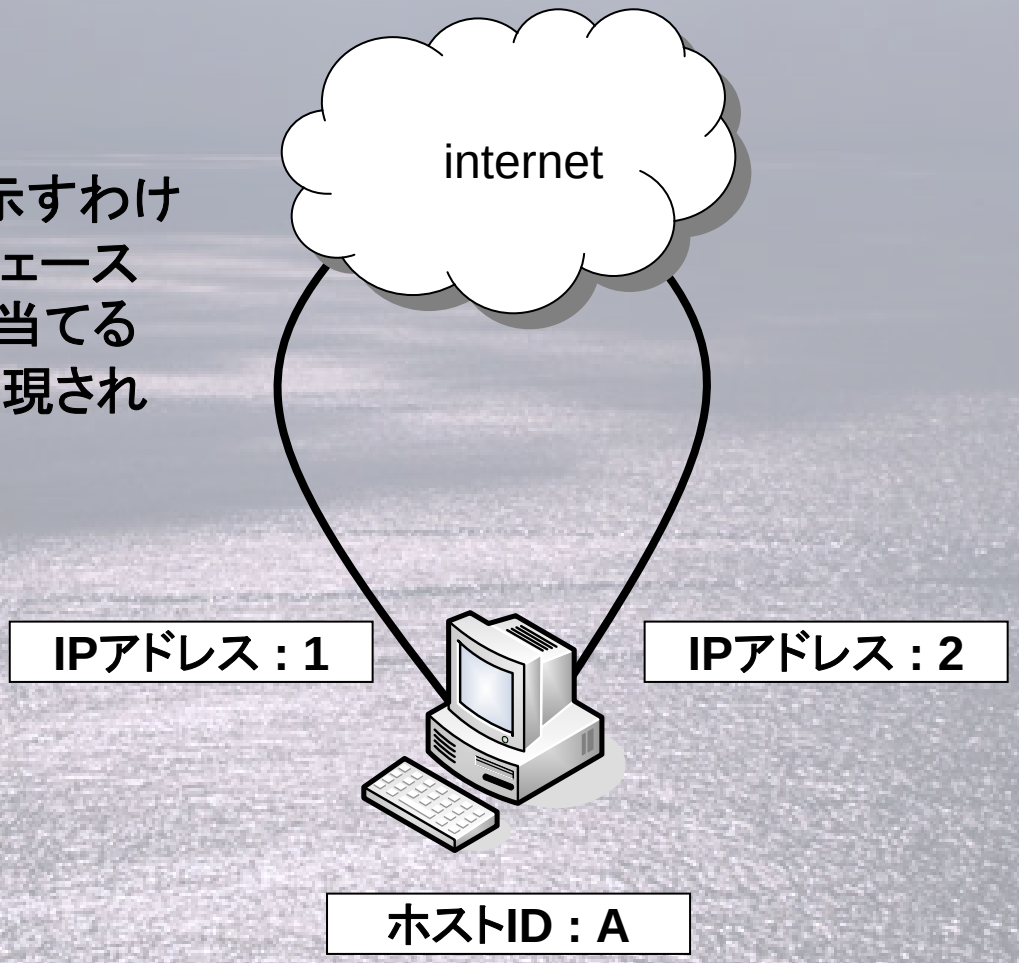
ホストID＝公開鍵

ホストIDタグ (128bit)

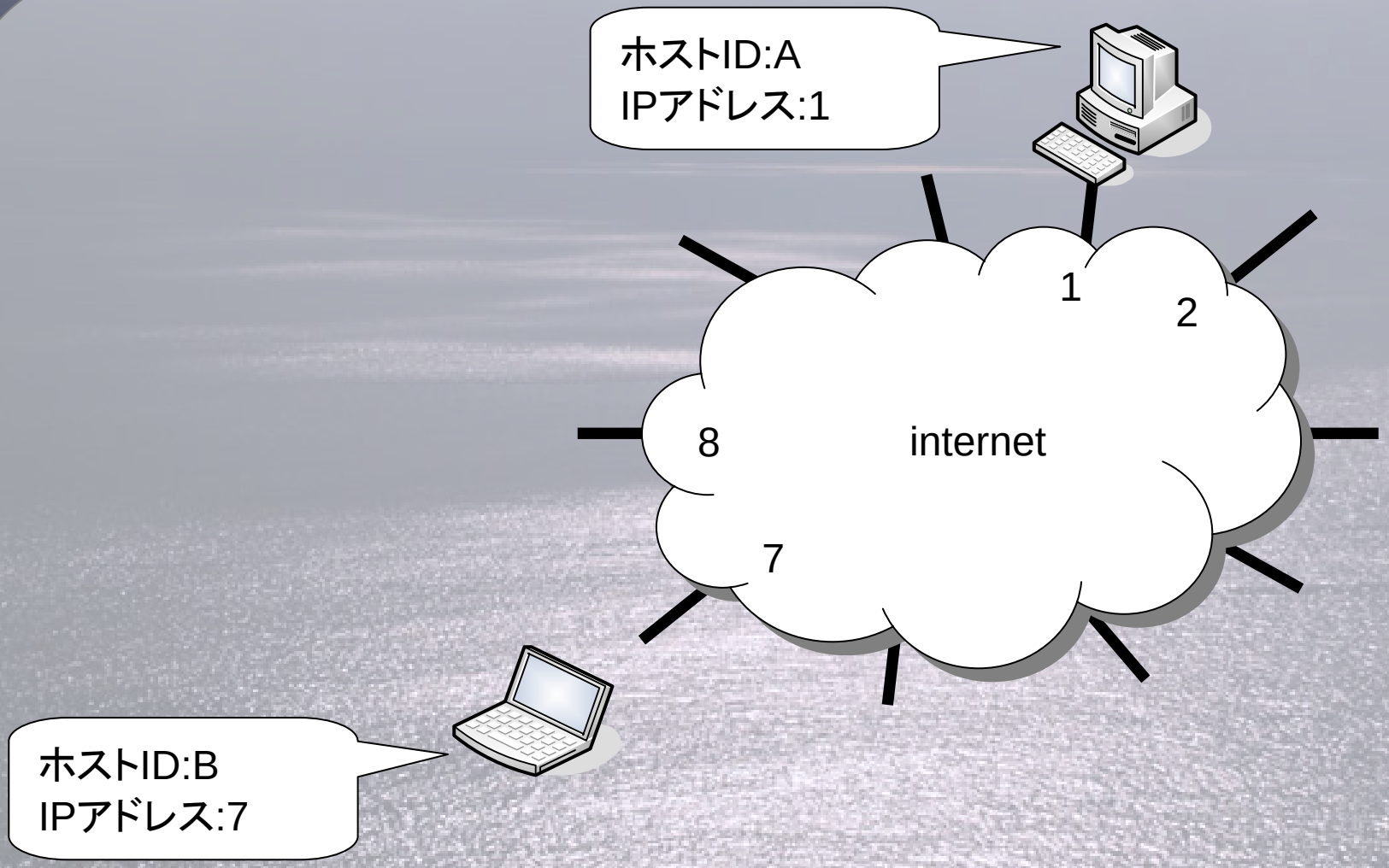
プロセスはこれを用いる

マルチホーミングの実現

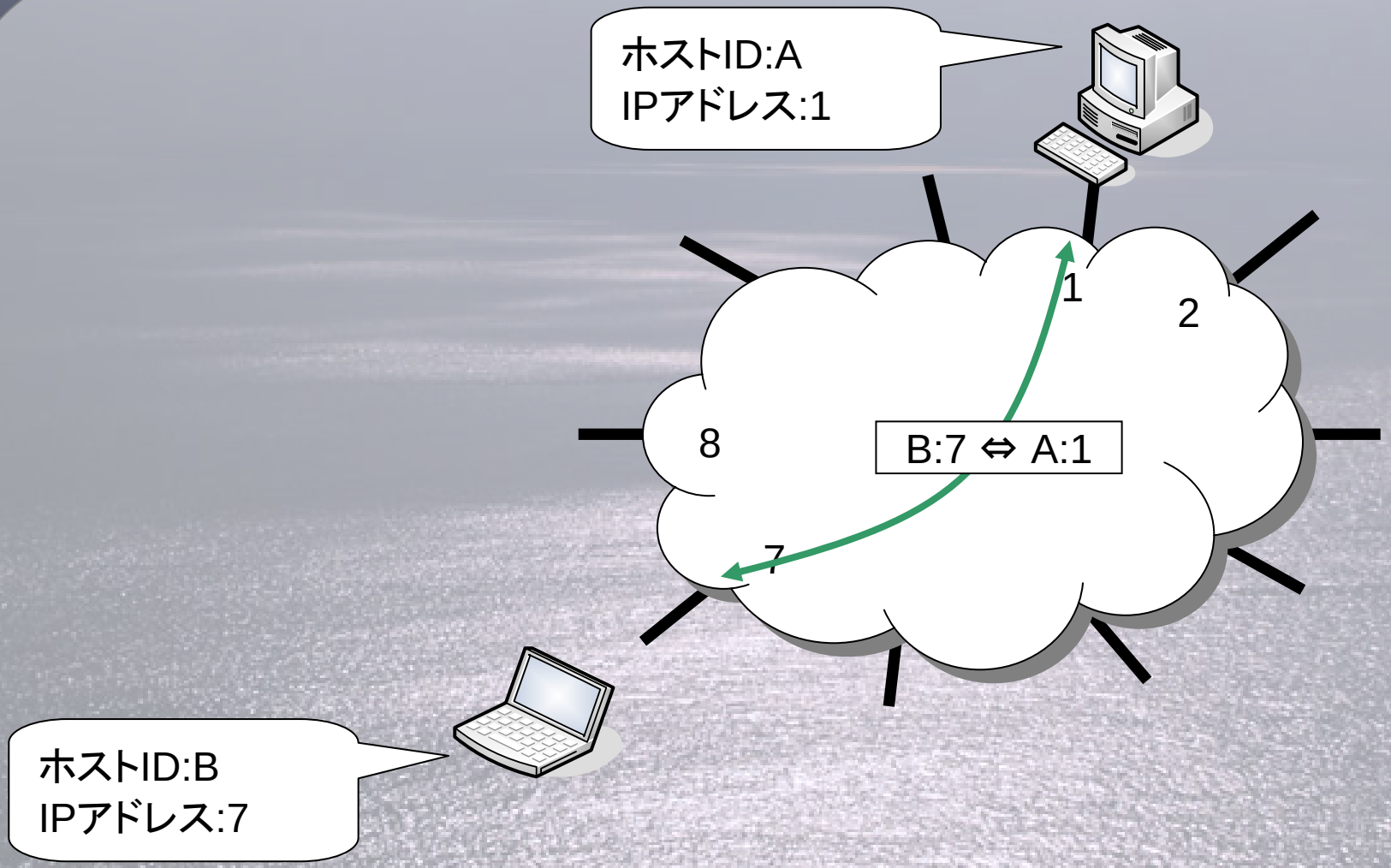
アドレスがホストそのものを示すわけではないので複数のインタフェースにそれぞれIPアドレスを割り当てるだけでマルチホーミングが実現される.



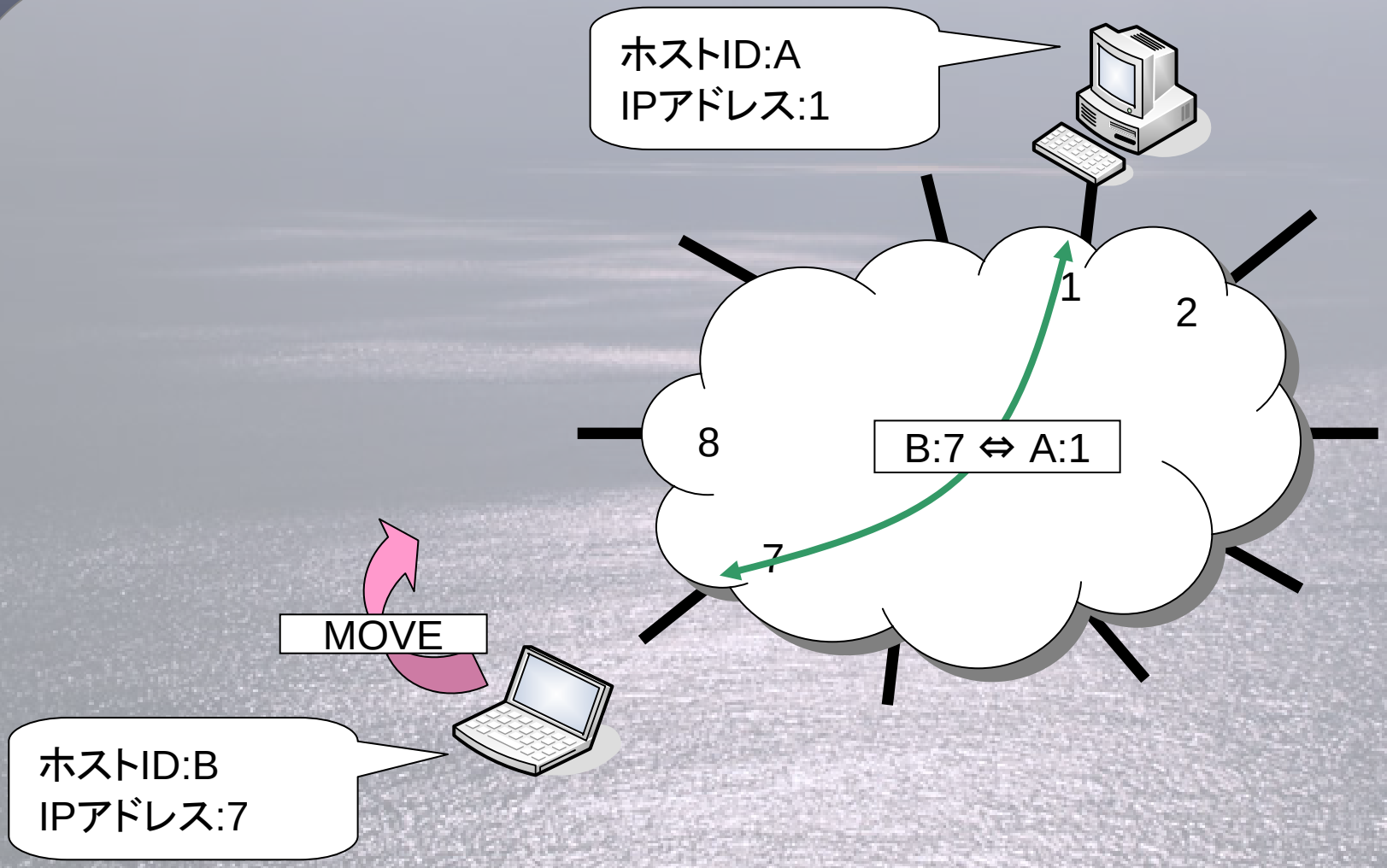
モビリティの実現



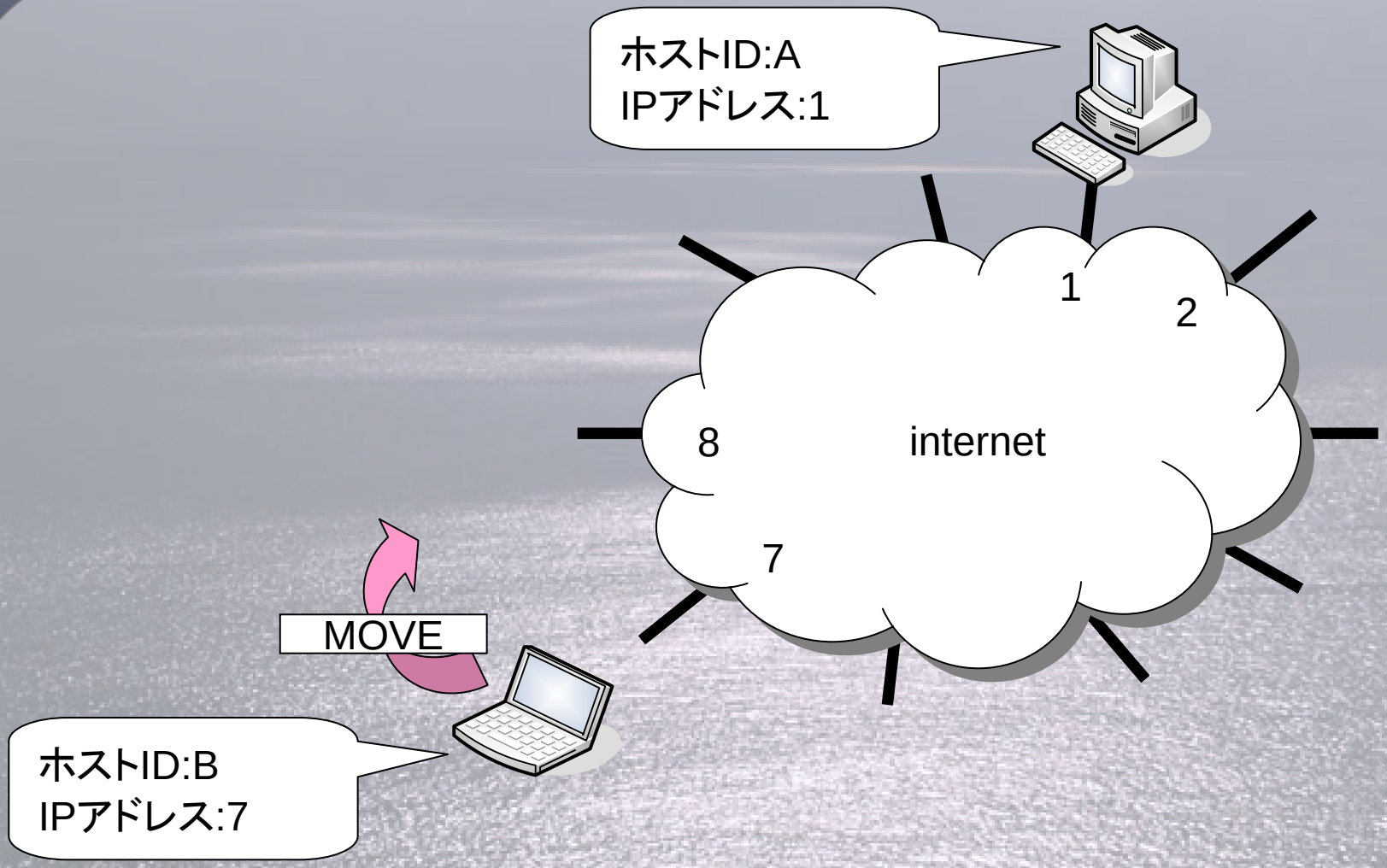
モビリティの実現



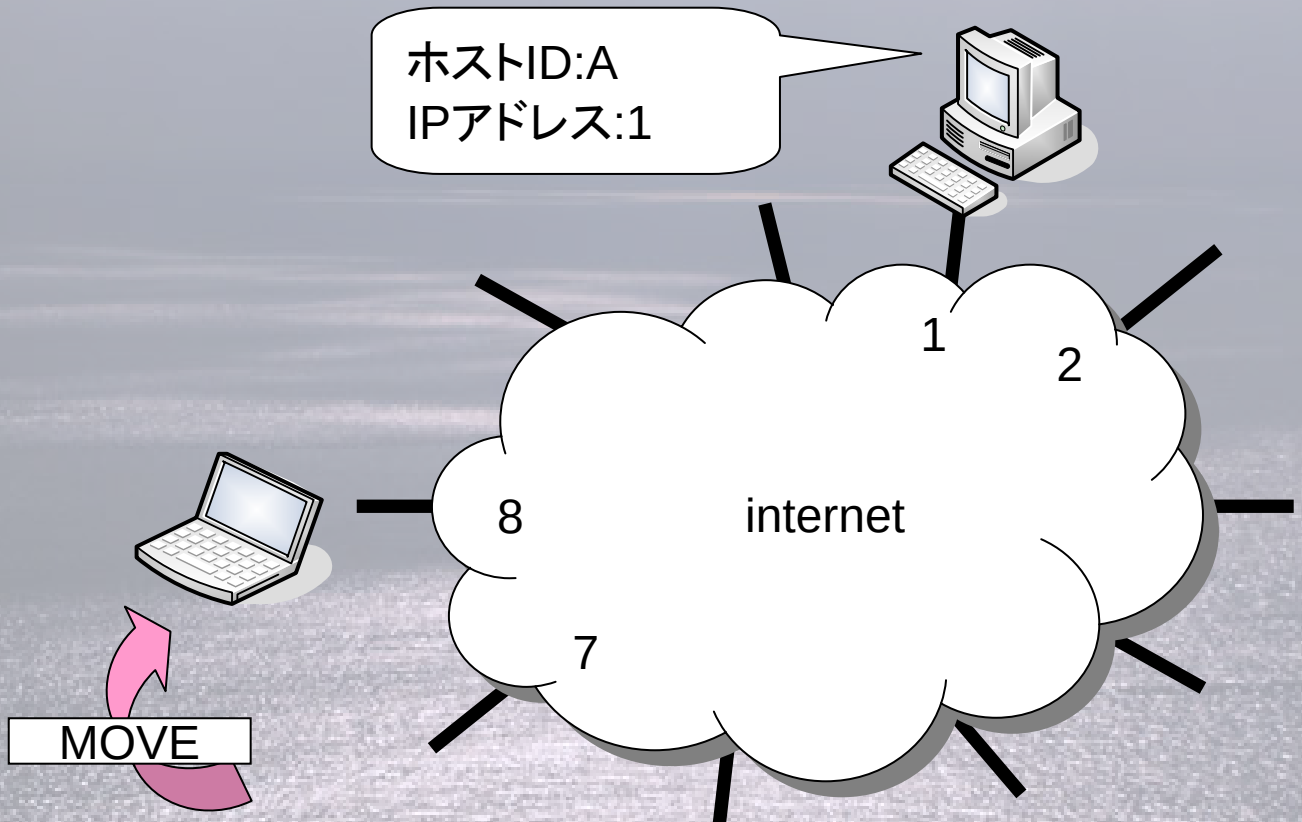
モビリティの実現



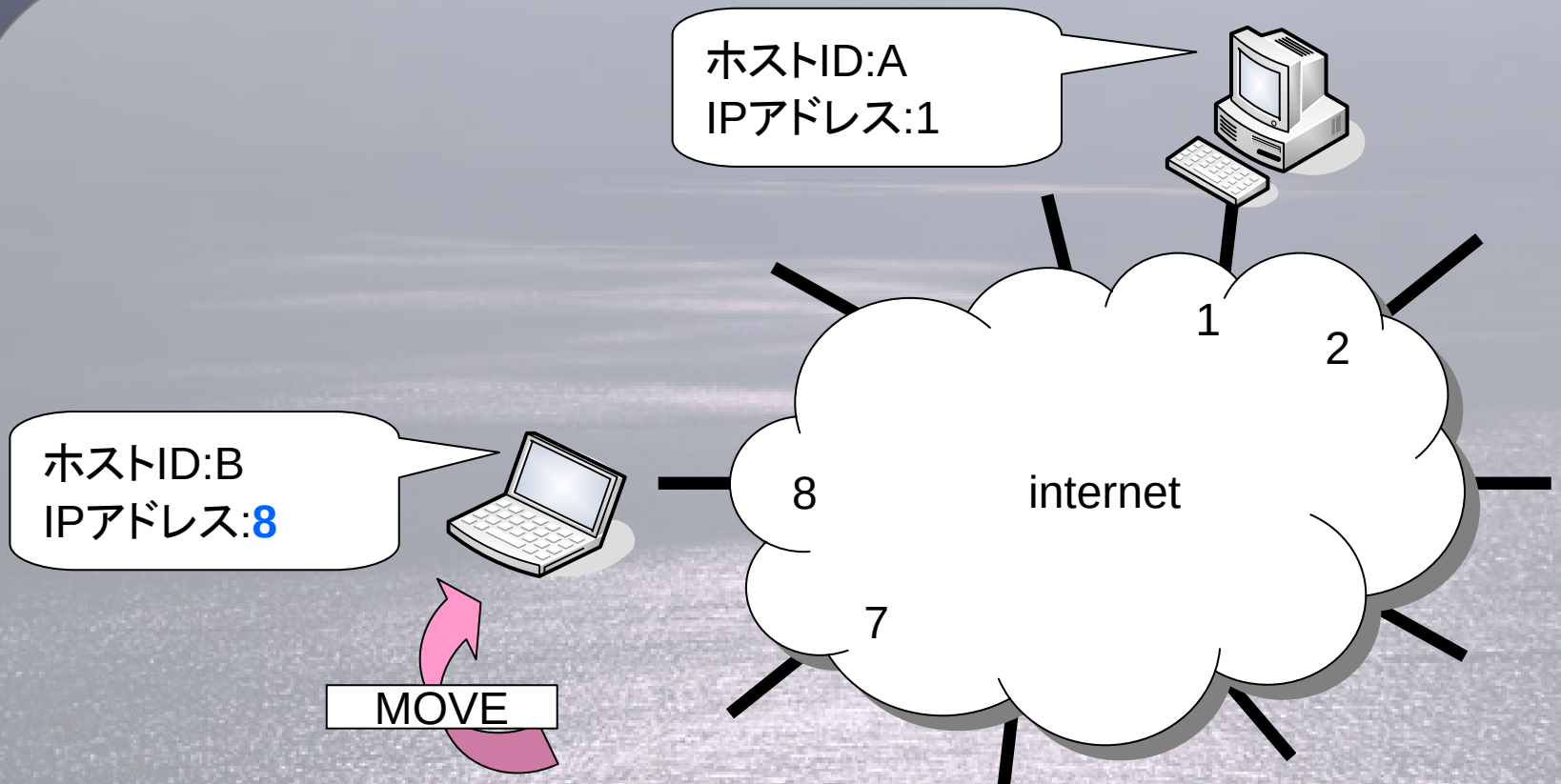
モビリティの実現



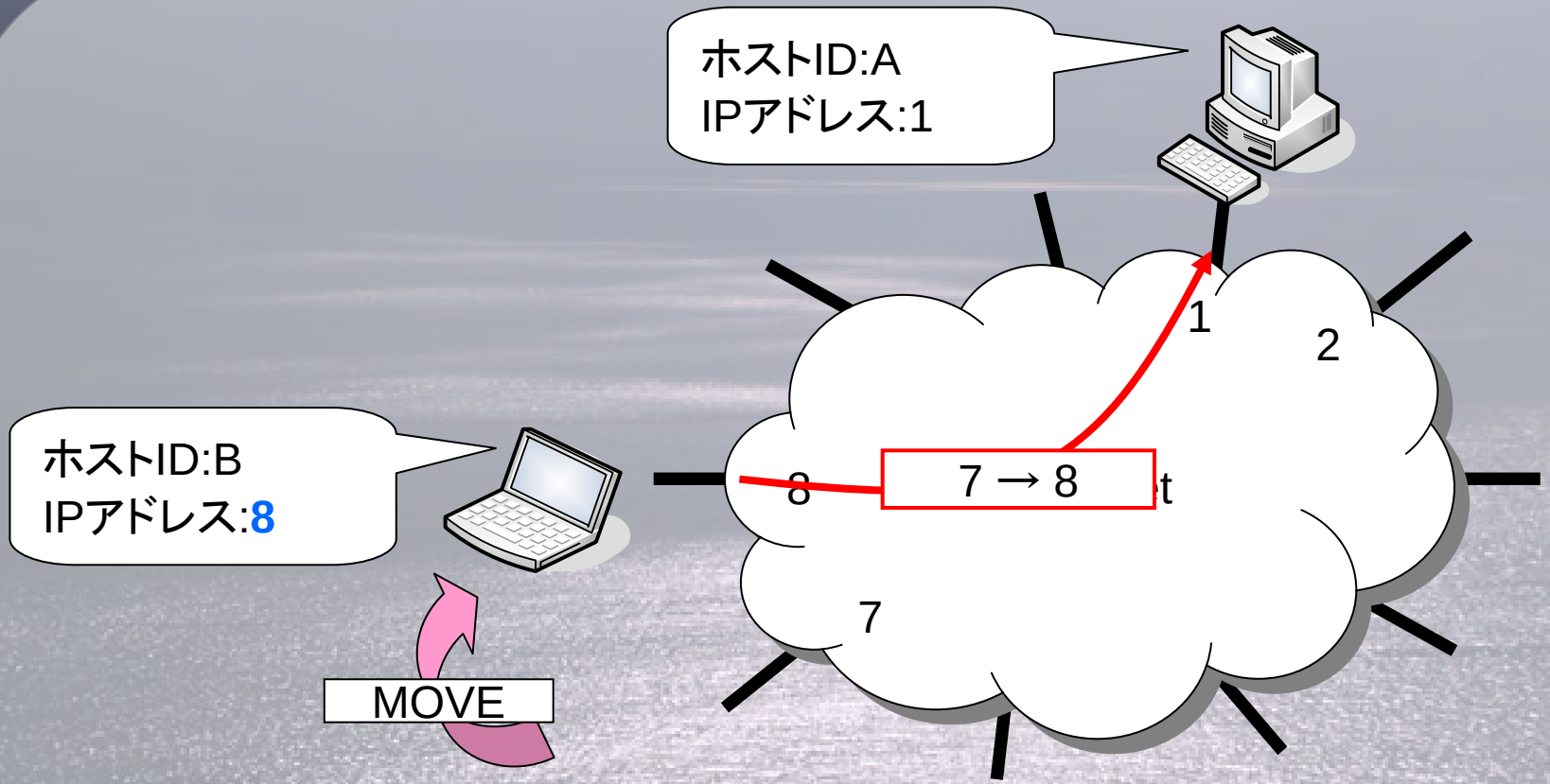
モビリティの実現



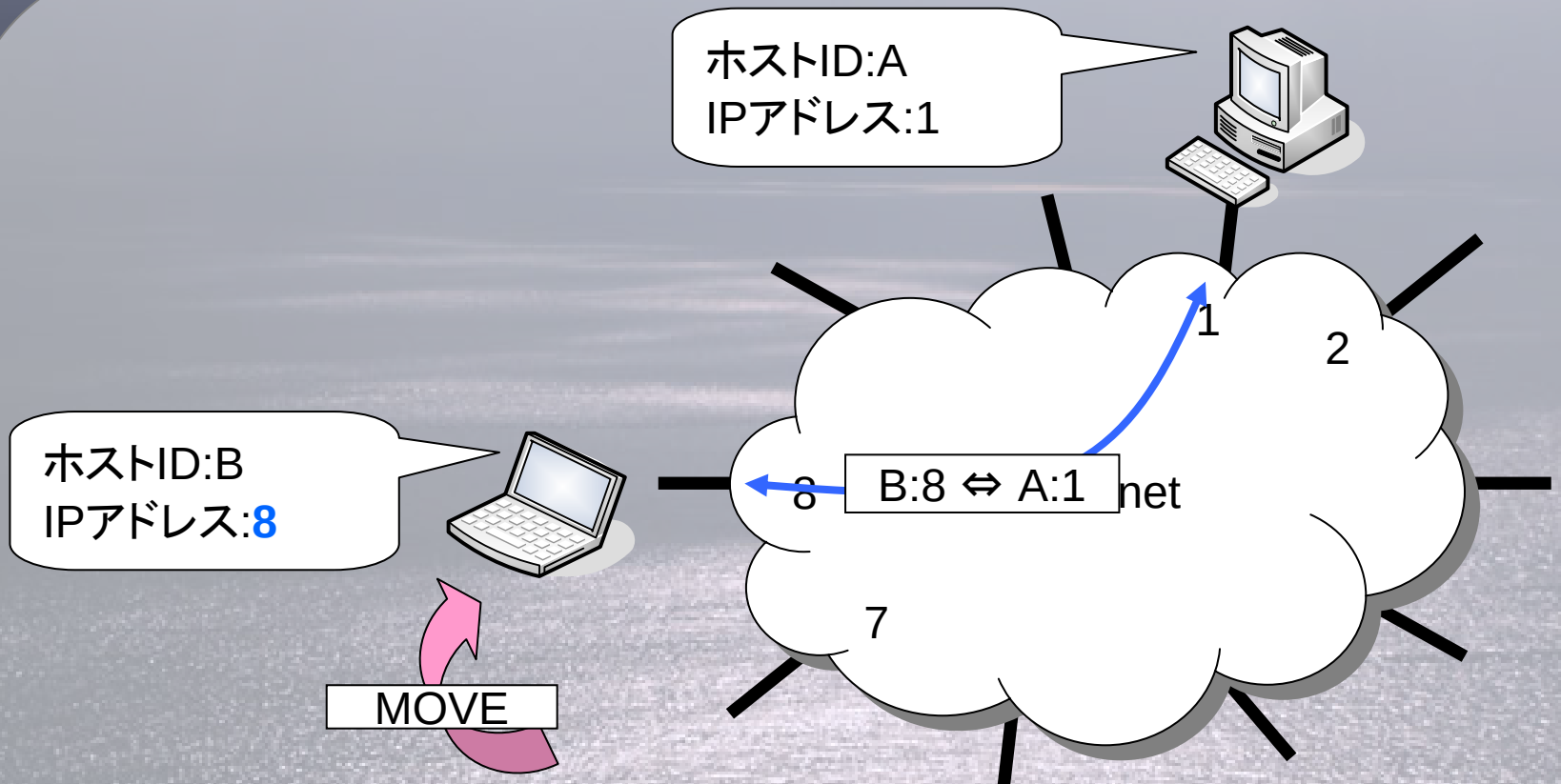
モビリティの実現



モビリティの実現

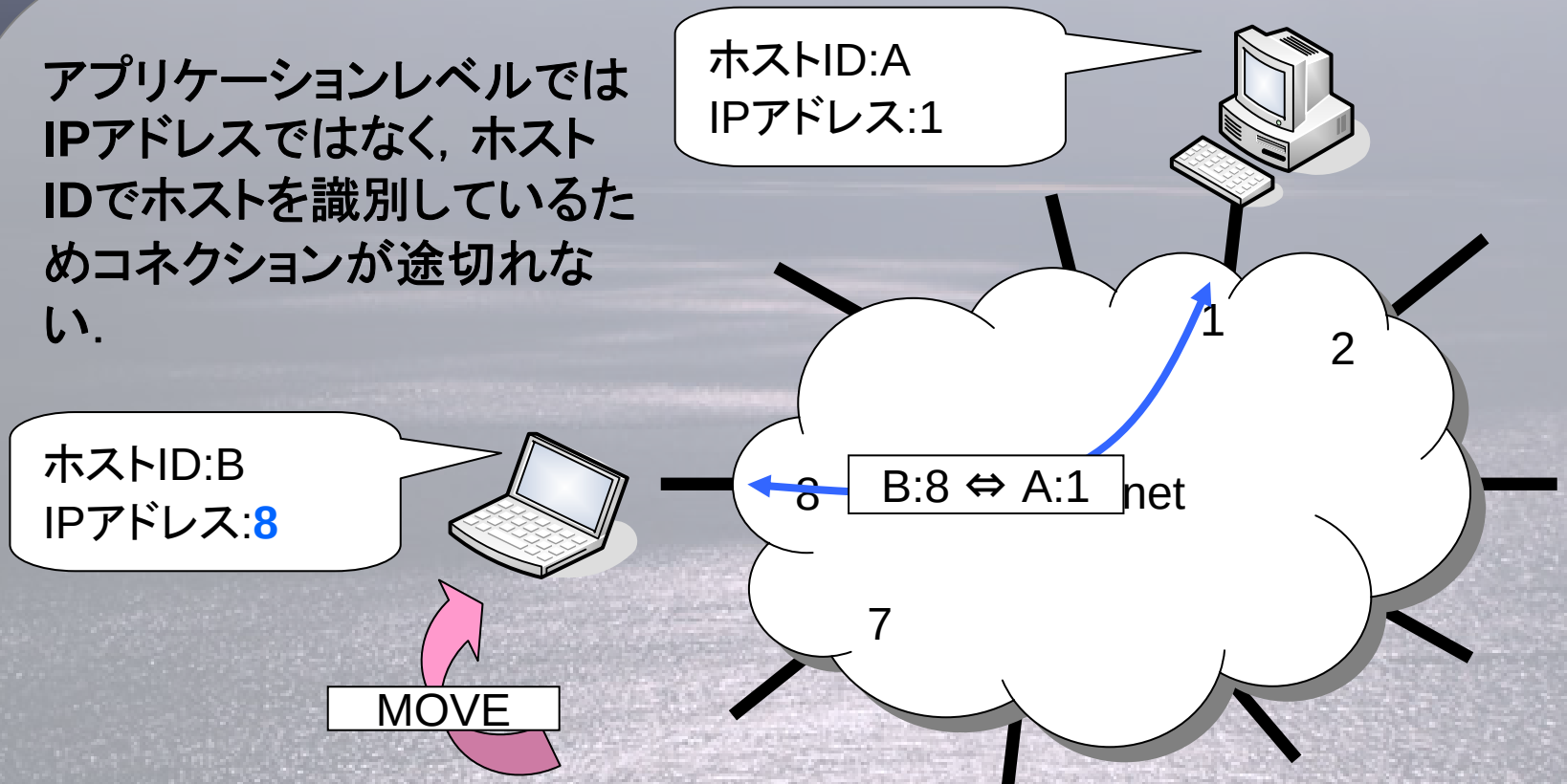


モビリティの実現



モビリティの実現

アプリケーションレベルではIPアドレスではなく，ホストIDでホストを識別しているためコネクションが途切れない．



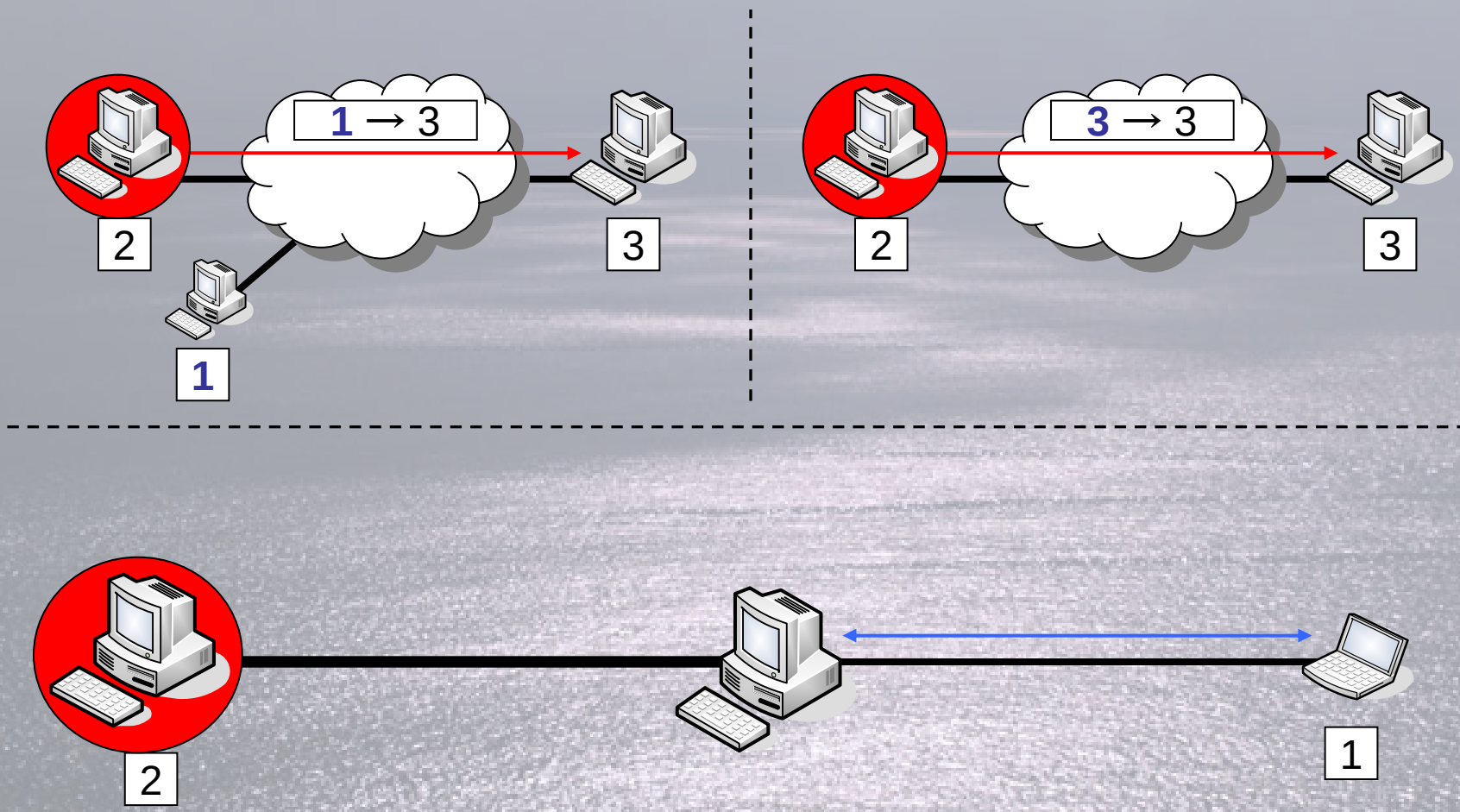
セキュリティ

- IPアドレスは位置の情報のみを示す
- 移動に伴って，IPアドレスは可換に使用する

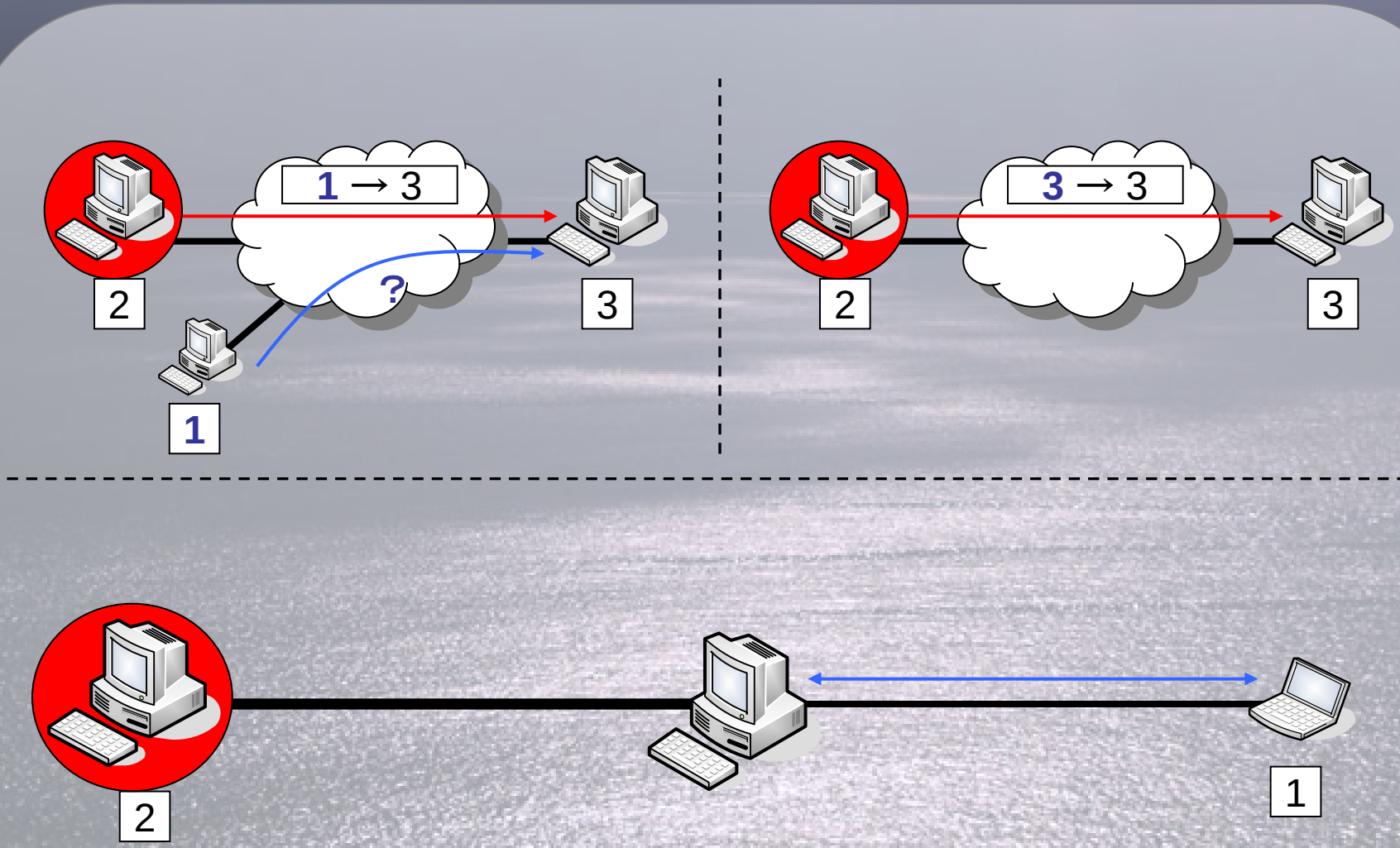
Address Stealing攻撃

Address Flooding攻撃

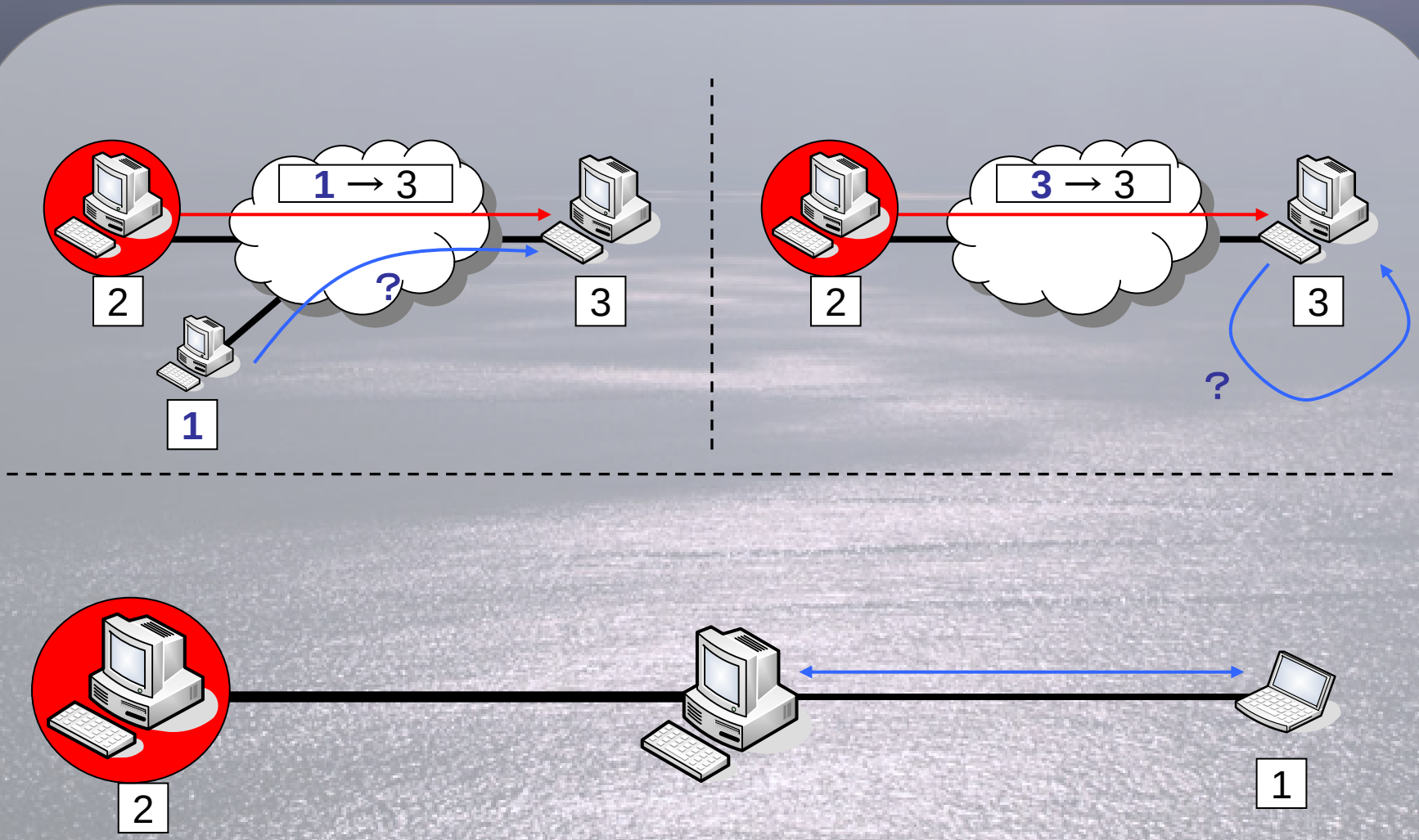
Address Stealing攻撃



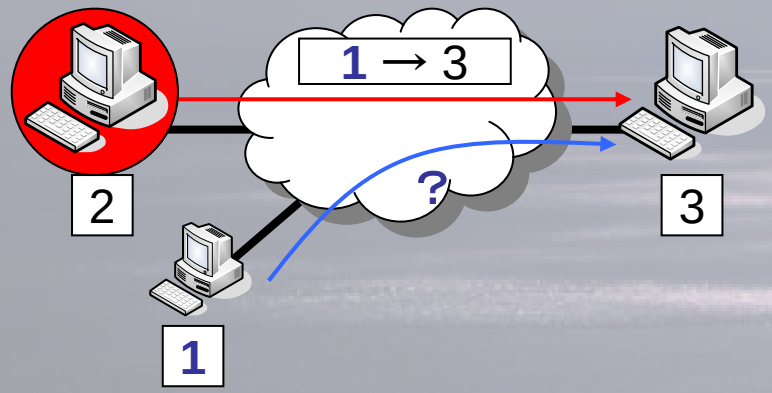
Address Stealing攻撃

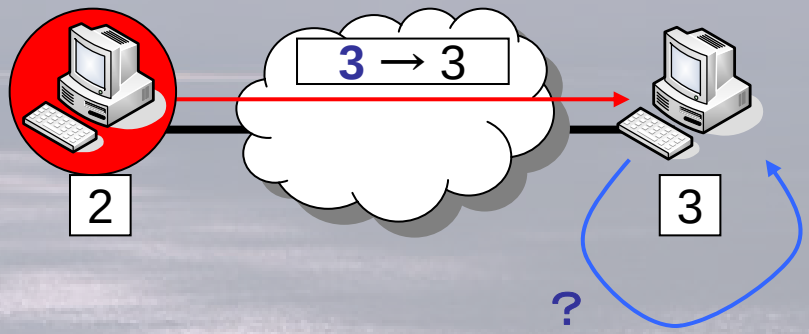


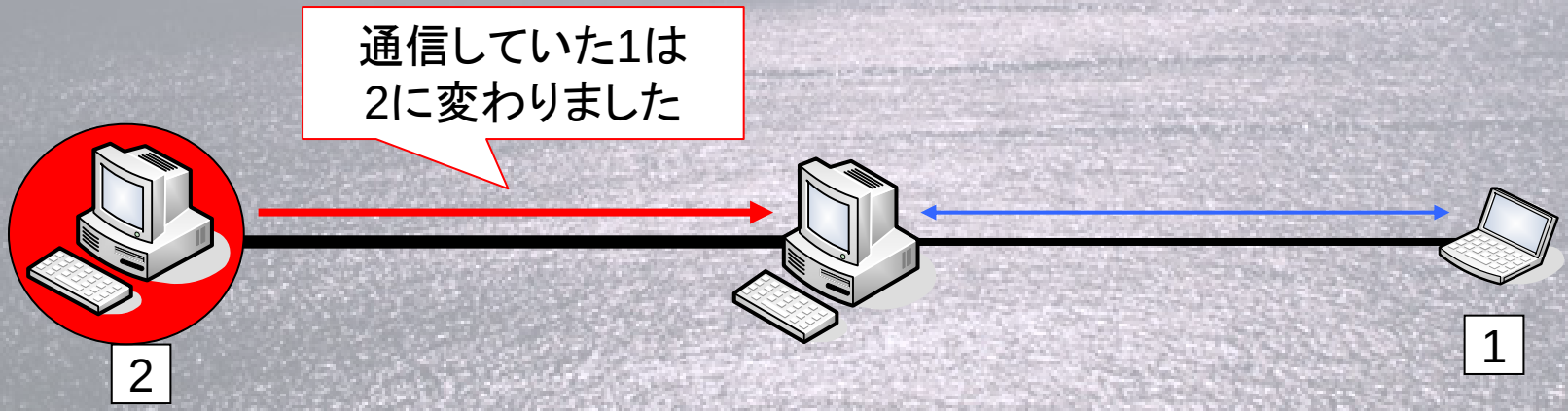
Address Stealing攻撃



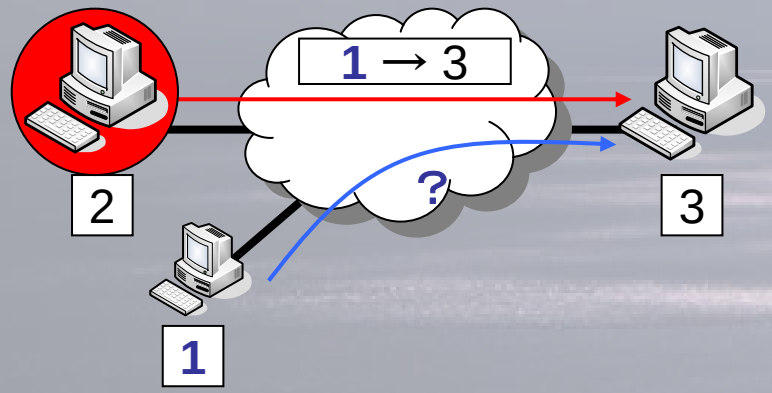
Address Stealing攻撃

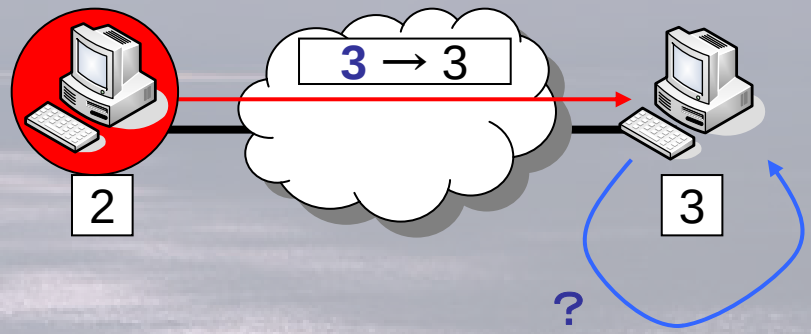


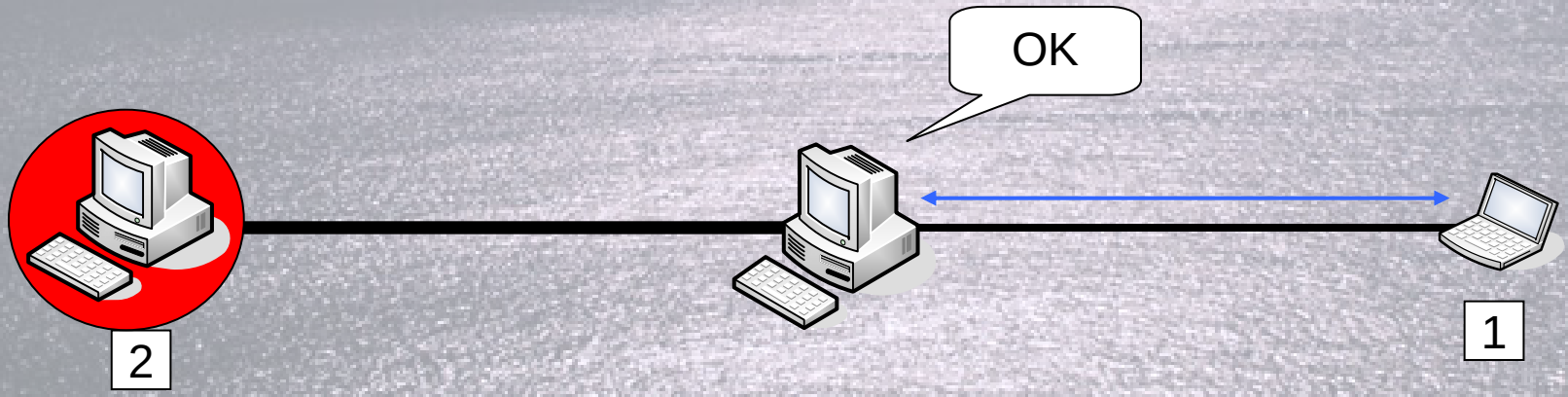




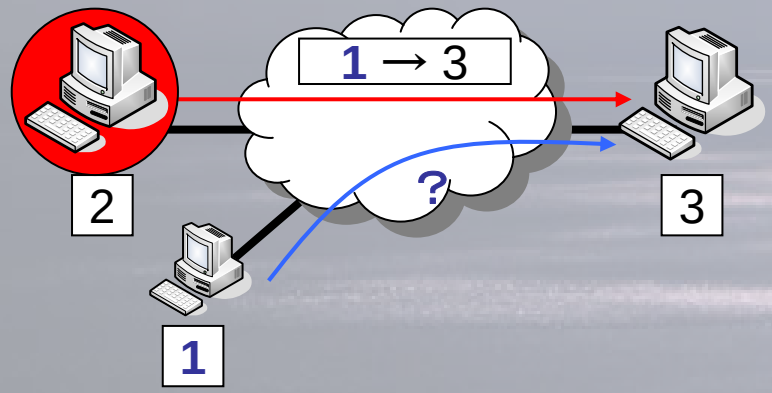
Address Stealing攻撃

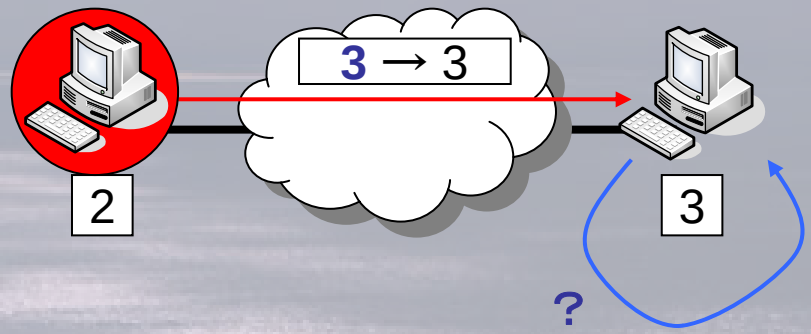


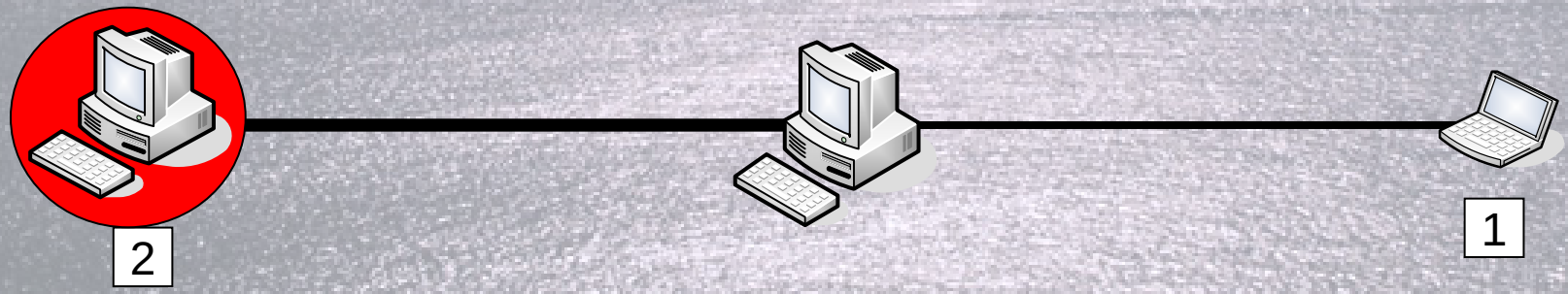




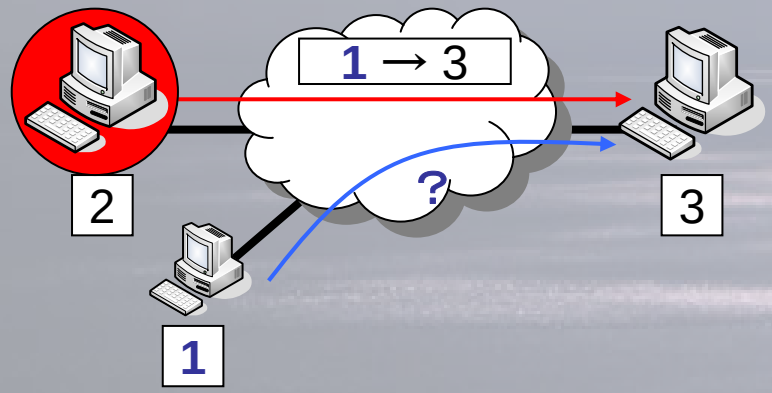
Address Stealing攻撃

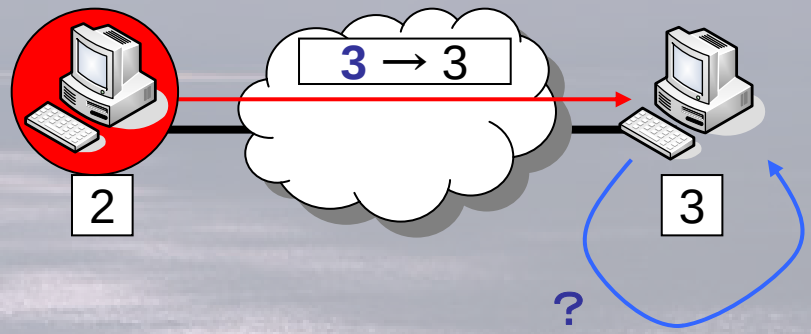


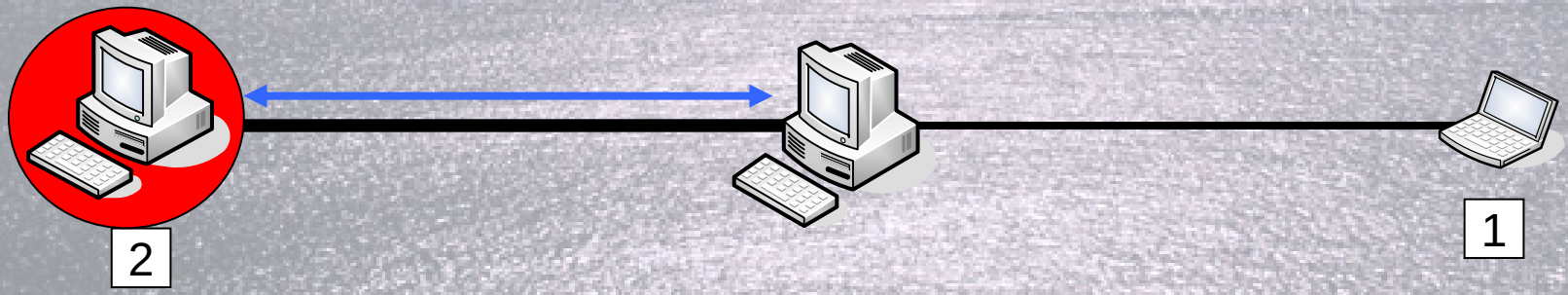




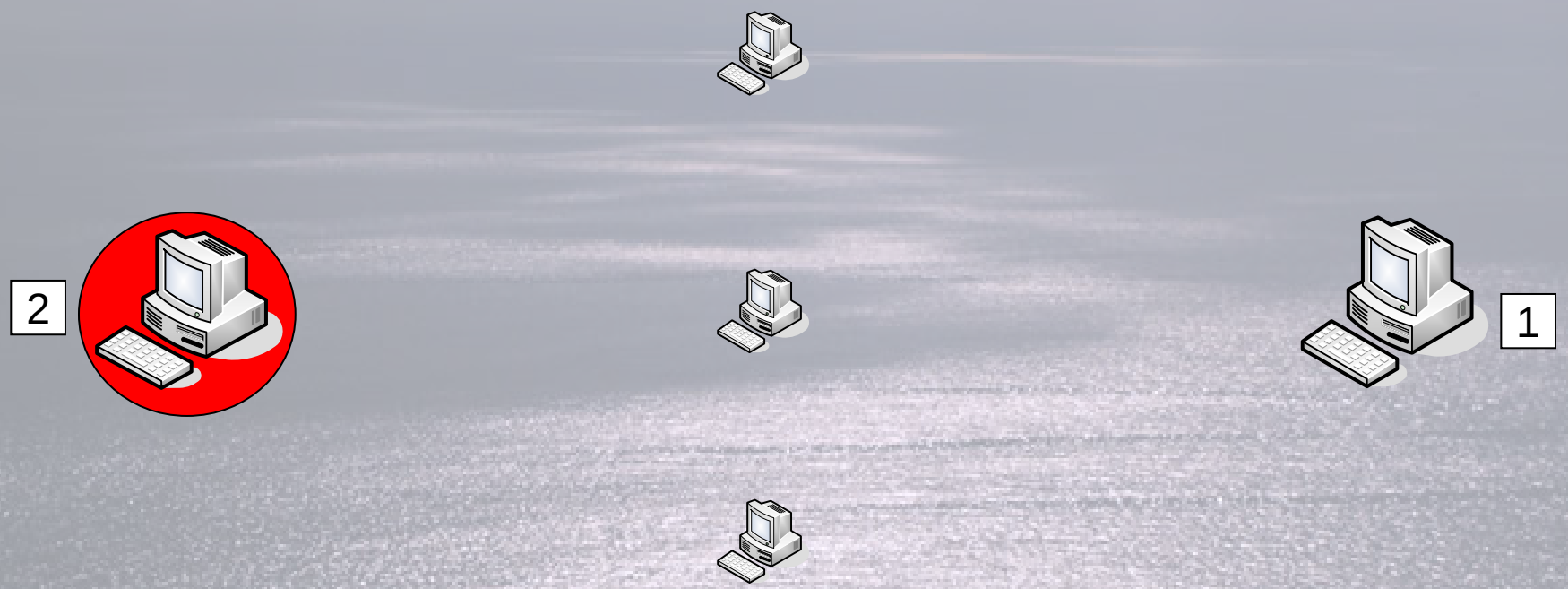
Address Stealing攻撃



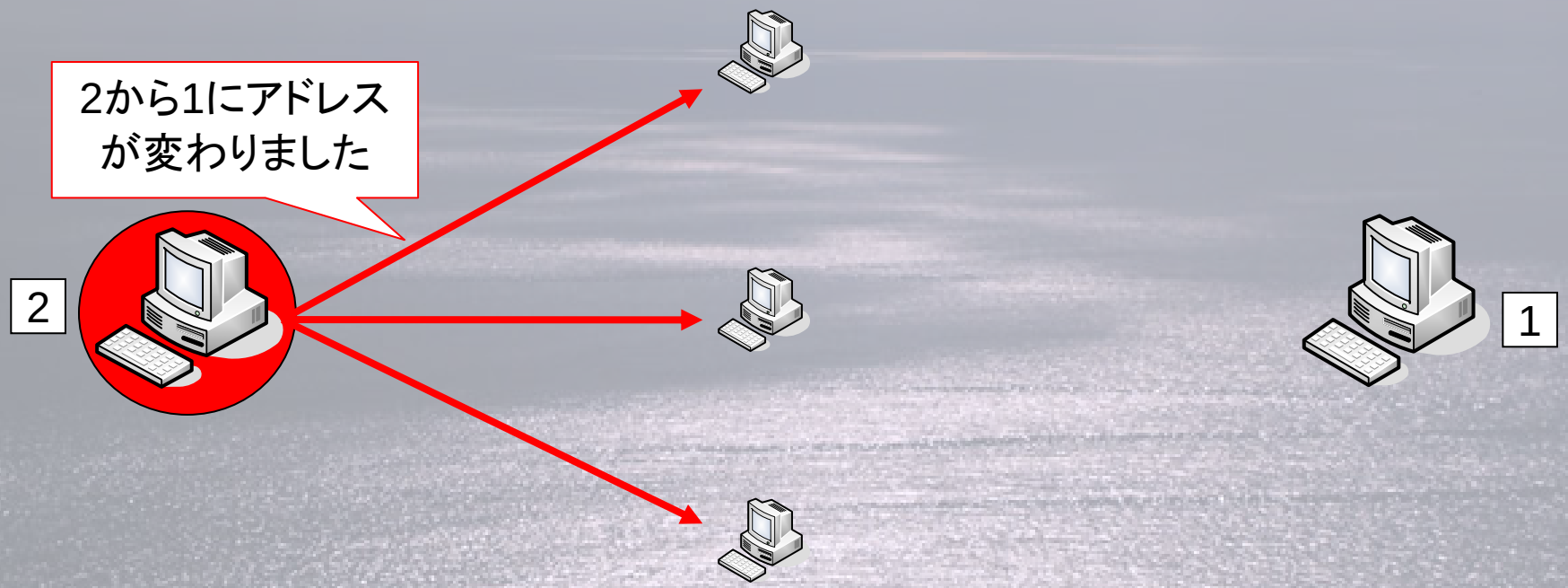




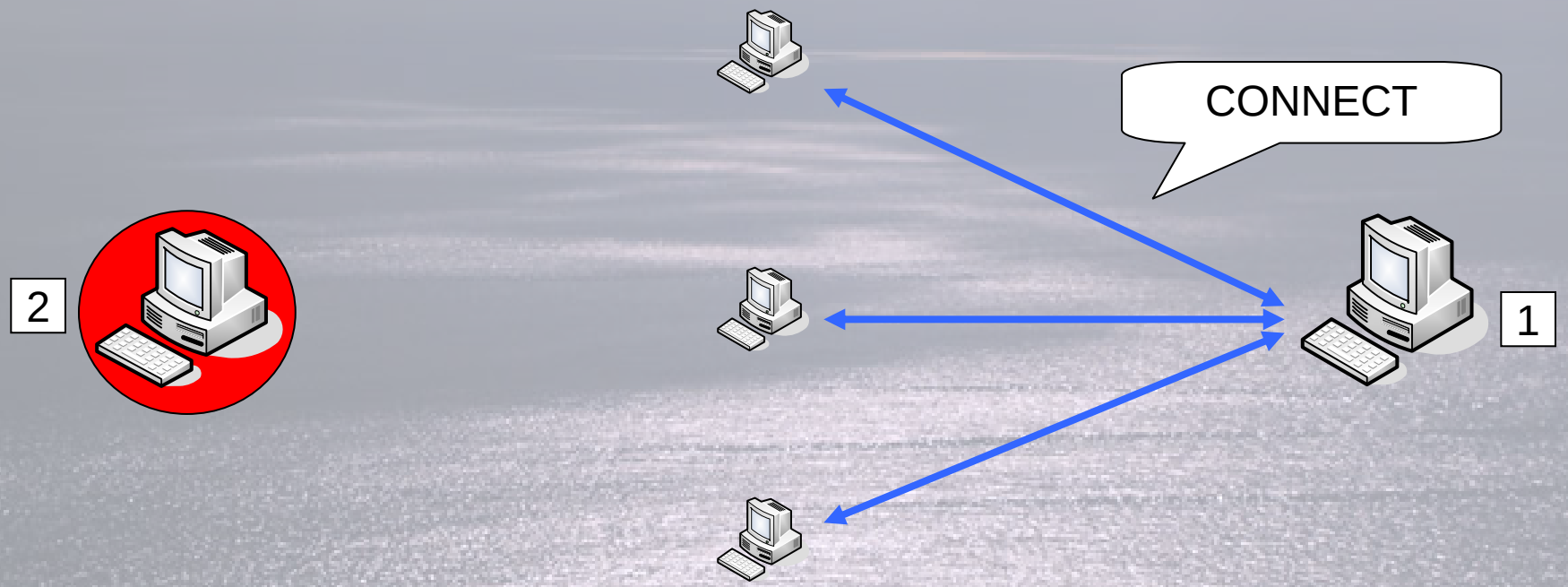
Address Flooding攻撃



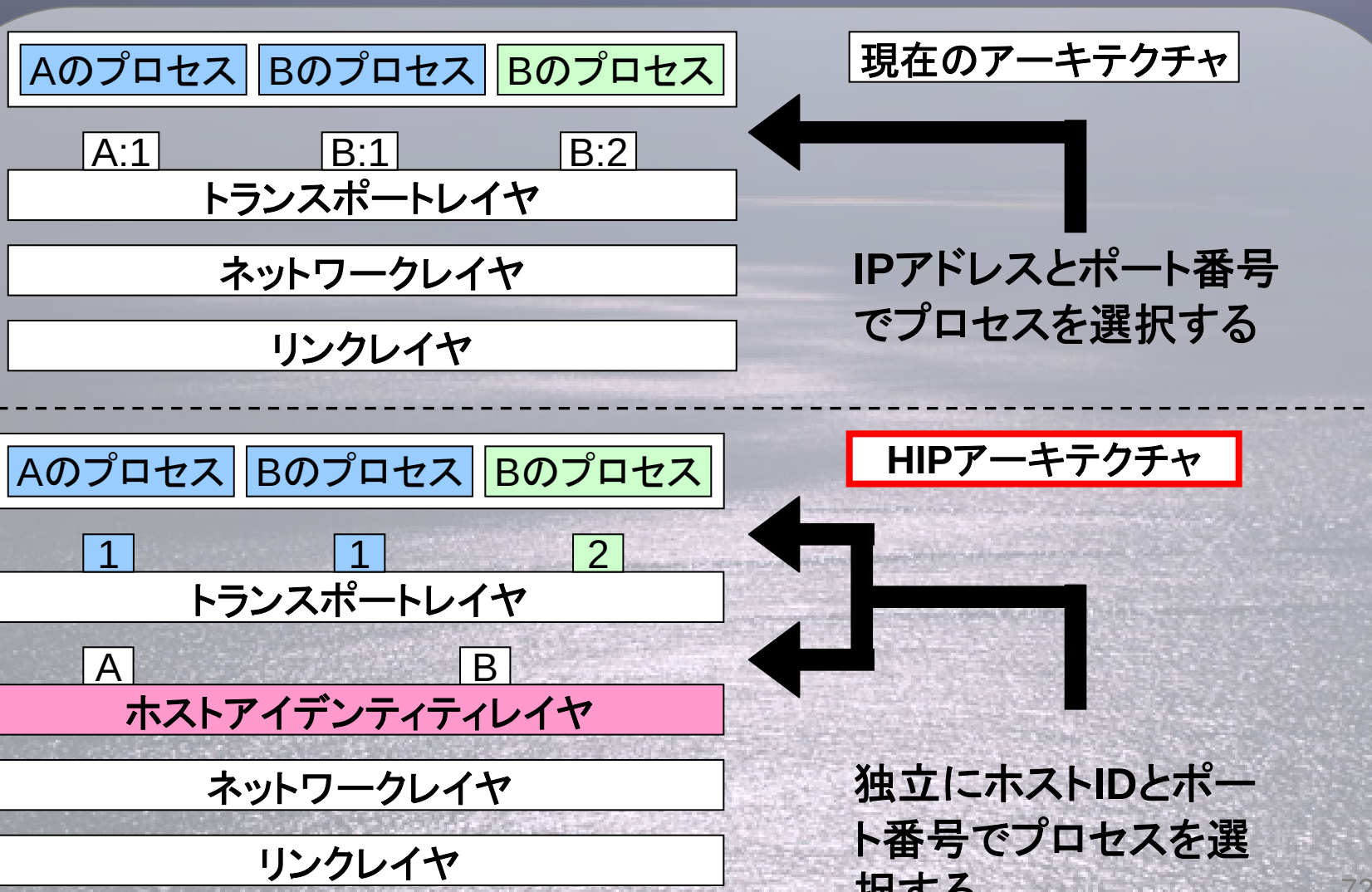
Address Flooding攻撃



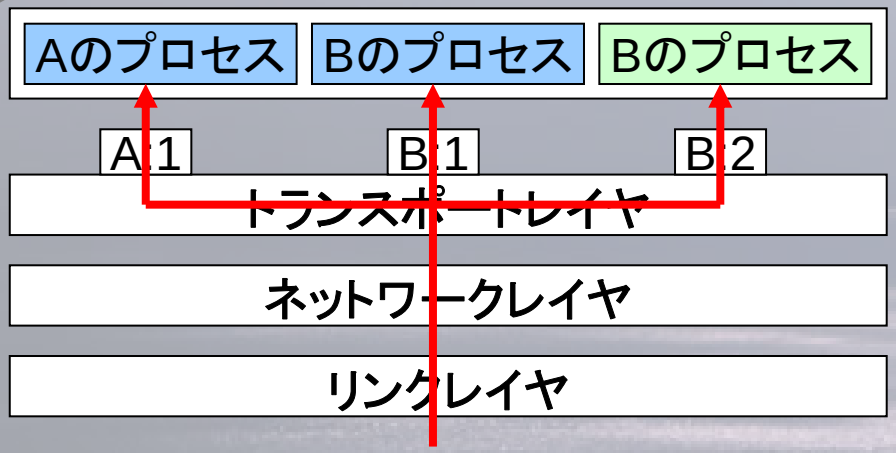
Address Flooding攻撃



HIPアーキテクチャ

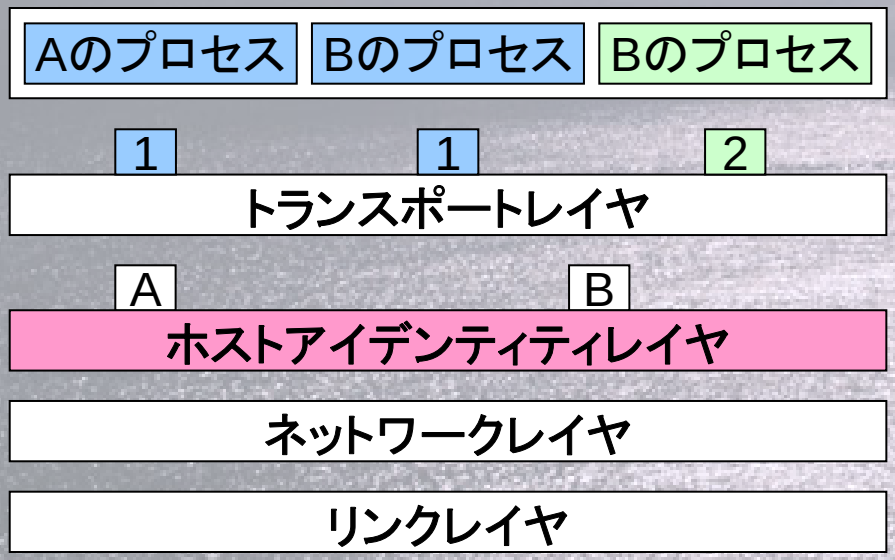

Aのプロセス
Bのプロセス
Bのプロセス
A:1
B:1
B:2
トランスポートレイヤ
ネットワークレイヤ
リンクレイヤ
現在のアーキテクチャ
IPアドレスとポート番号でプロセスを選択する
Aのプロセス
Bのプロセス
Bのプロセス
1
1
2
トランスポートレイヤ
A
B
ホストアイデンティティレイヤ
ネットワークレイヤ
リンクレイヤ
HIPアーキテクチャ
独立にホストIDとポート番号でプロセスを選択する

HIPアーキテクチャ



現在のアーキテクチャ

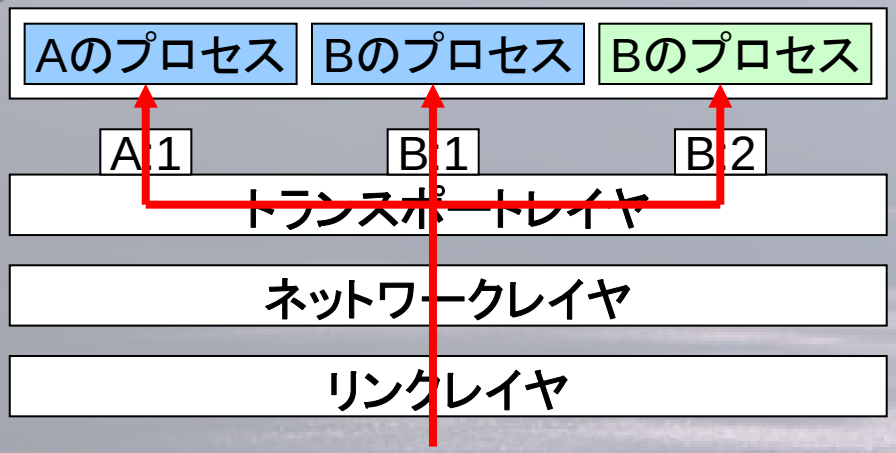
IPアドレスとポート番号でプロセスを選択する



HIPアーキテクチャ

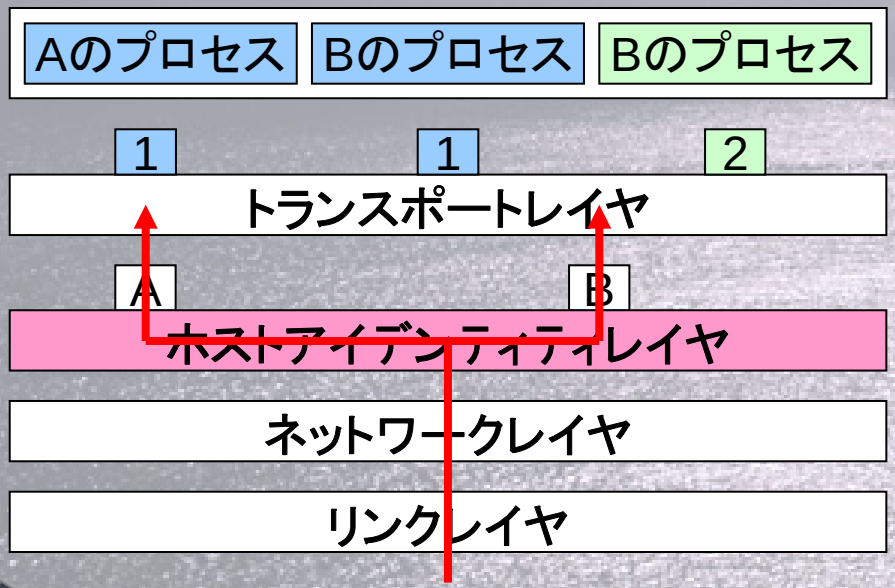
独立にホストIDとポート番号でプロセスを選択する

HIPアーキテクチャ



現在のアーキテクチャ

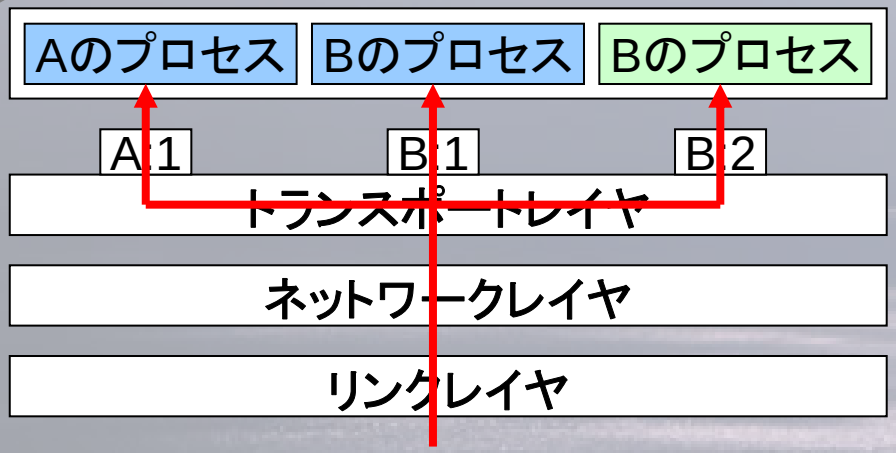
IPアドレスとポート番号でプロセスを選択する



HIPアーキテクチャ

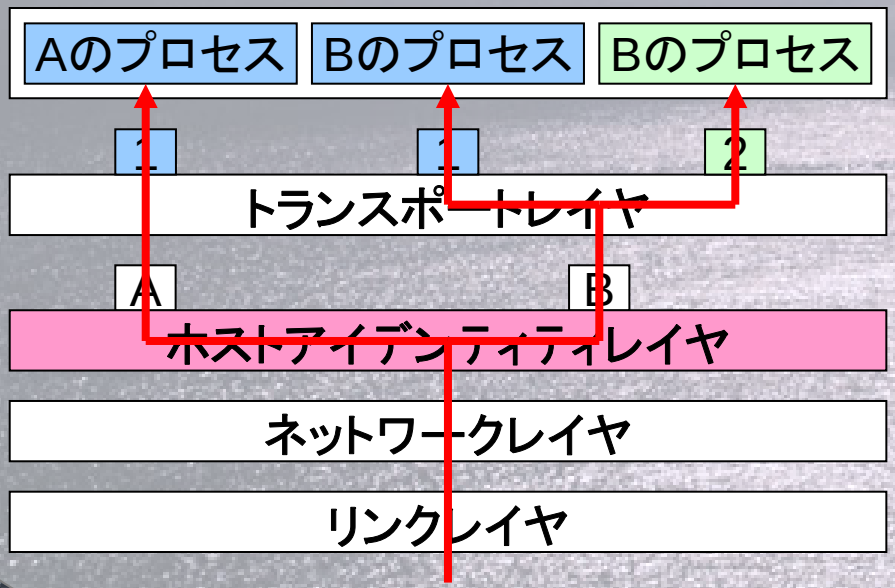
独立にホストIDとポート番号でプロセスを選択する

HIPアーキテクチャ



現在のアーキテクチャ

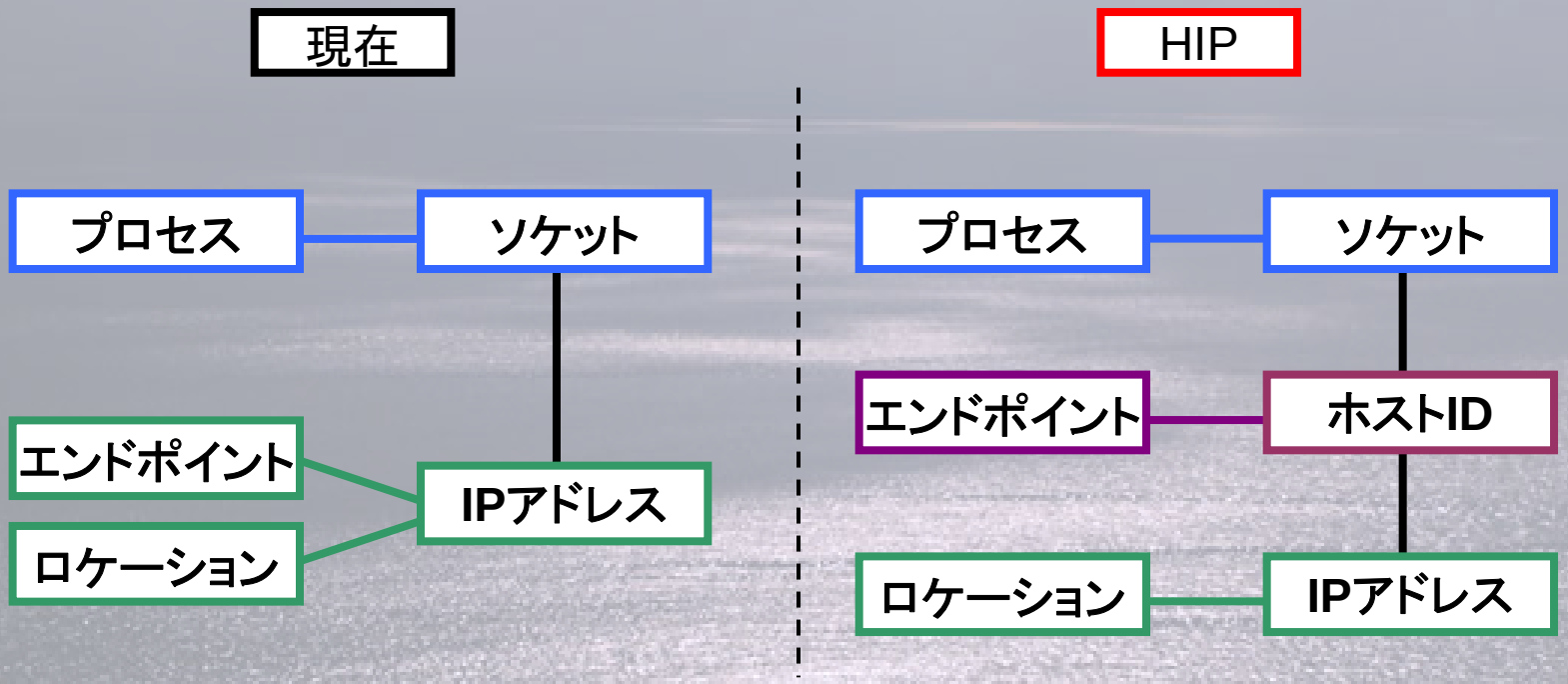
IPアドレスとポート番号でプロセスを選択する



HIPアーキテクチャ

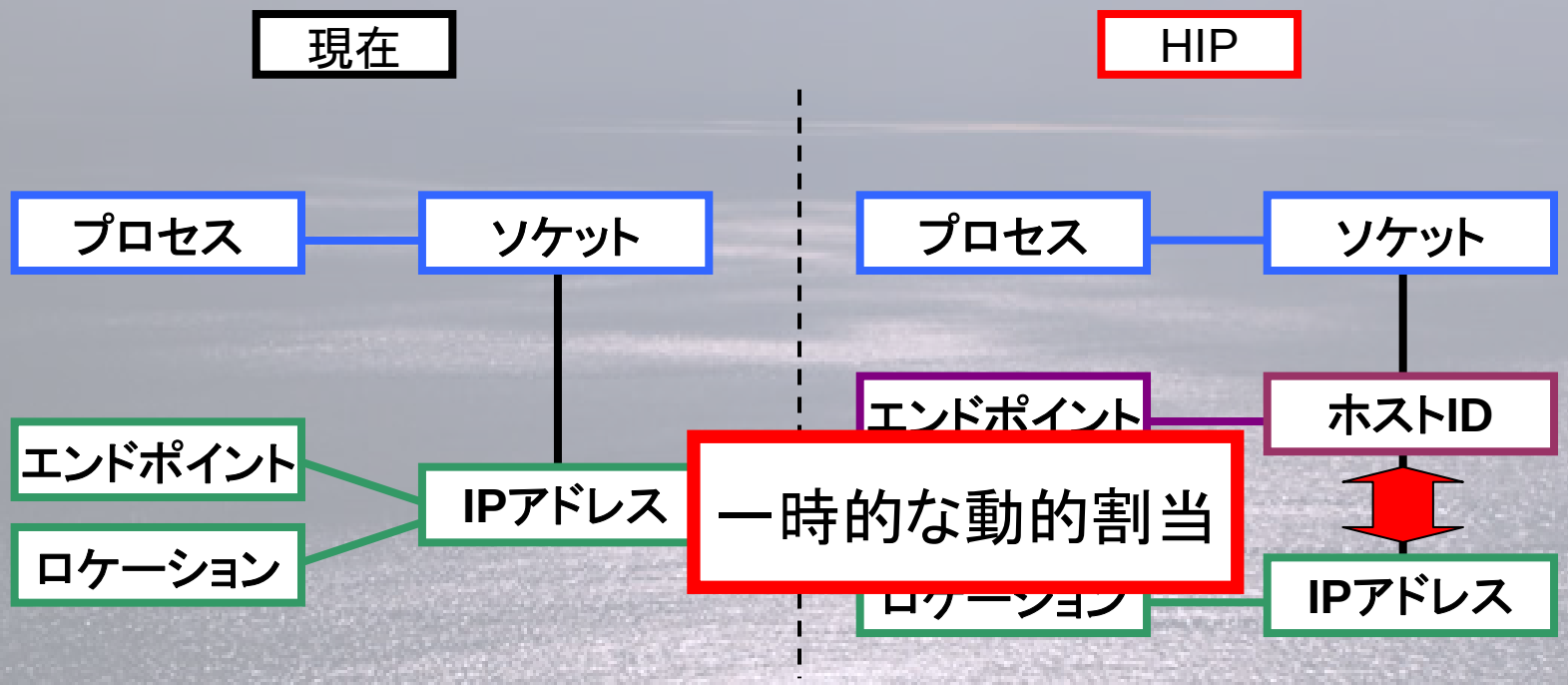
独立にホストIDとポート番号でプロセスを選択する

HIPアーキテクチャ



パケットにはホストID情報が必要

HIPアーキテクチャ



パケットにはホストID情報が必要

HIPパケット

HIP:自身と通信相手のホストID＝公開鍵

Address Stealing攻撃対策

HIPパケットはESPで保護されるため

・守秘性

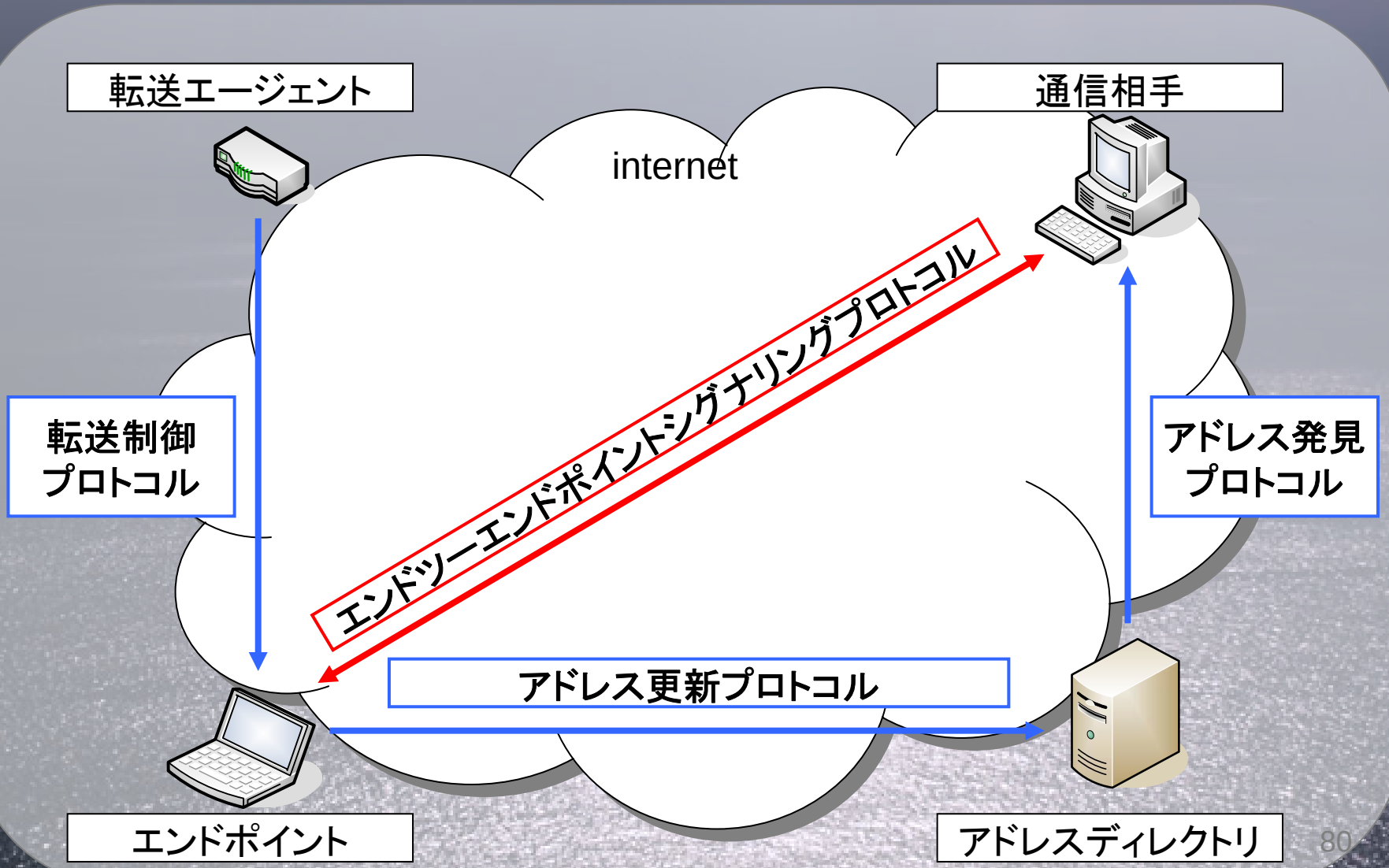
・データ生成元の認証

が約束される.

HIPアーキテクチャ構成プロトコル

- アドレス発見プロトコル
- エンドツーエンドポイントシグナリングプロトコル
- アドレス更新プロトコル
- 転送制御プロトコル

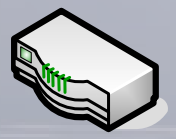
HIPアーキテクチャ構成プロトコル



エンドツーエンドポイントシグナリングプロトコル

パケット転送エージェント

転送テーブル
101 → 4
102 → 5
103 → -



転送要請はホストID:A
とペアの秘密鍵で署名
する

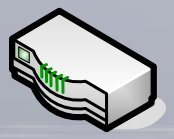
ホストID:A

1

2

パケット転送エージェント

転送テーブル
101 → 4
102 → 5
103 → -



転送要請はホストID:Aとペアの秘密鍵で署名する

転送要請

2へ転送してください

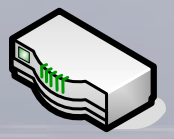
ホストID:A

1

2

パケット転送エージェント

転送テーブル
101 → 4
102 → 5
103 → 2



転送要請はホストID:Aとペアの秘密鍵で署名する

転送要請

2へ転送してください

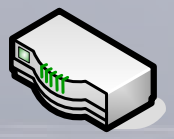
ホストID:A

1

2

パケット転送エージェント

転送テーブル
101 → 4
102 → 5
103 → 2



転送要請はホストID:Aとペアの秘密鍵で署名する

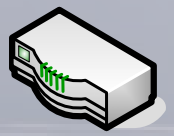
ホストID:A

1

2

パケット転送エージェント

転送テーブル
101 → 4
102 → 5
103 → 2



転送要請はホストID:Aとペアの秘密鍵で署名する

MOVE

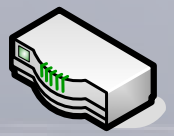
ホストID:A

1

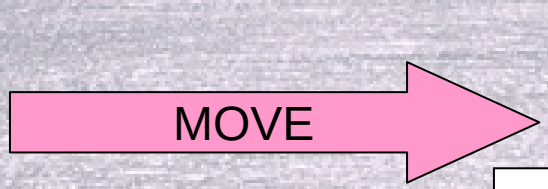
2

パケット転送エージェント

転送テーブル
101 → 4
102 → 5
103 → 2



転送要請はホストID:Aとペアの秘密鍵で署名する



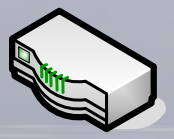
ホストID:A

1

2

パケット転送エージェント

転送テーブル
101 → 4
102 → 5
103 → 2



転送要請はホストID:Aとペアの秘密鍵で署名する

照合

MOVE

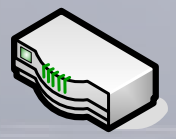
ホストID:A

1

2

パケット転送エージェント

転送テーブル
101 → 4
102 → 5
103 → 2



転送要請はホストID:Aとペアの秘密鍵で署名する

要求

MOVE

ホストID:A

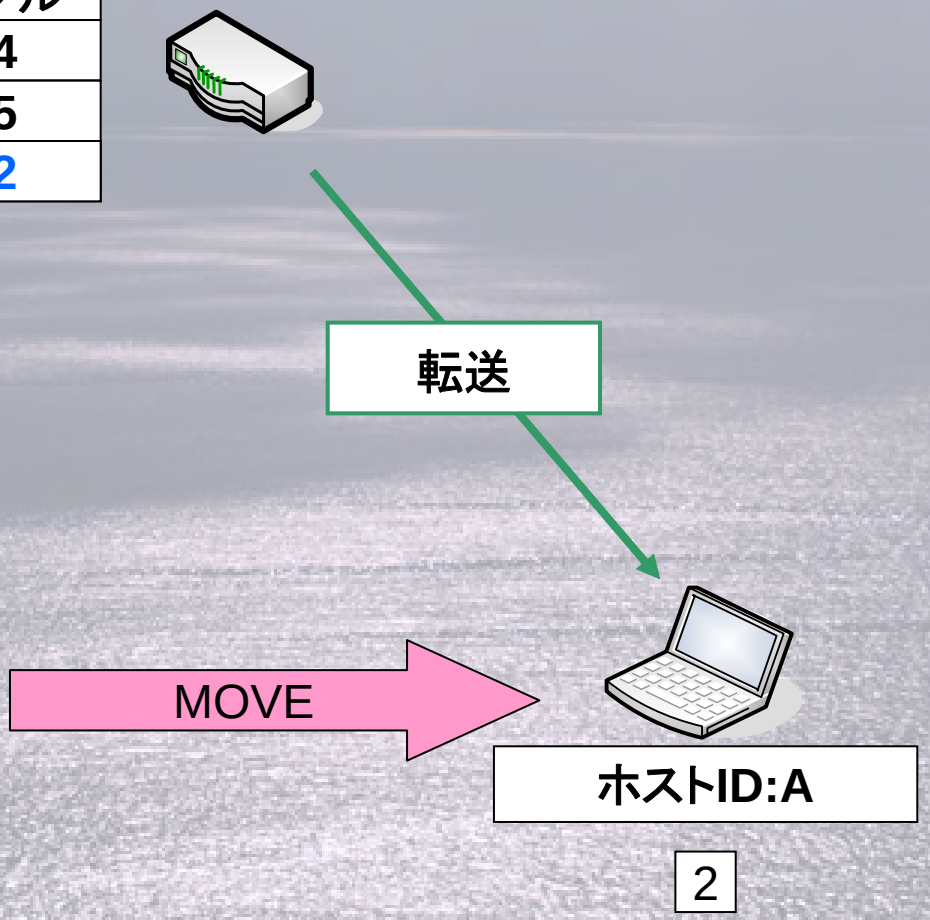
1

2

パケット転送エージェント

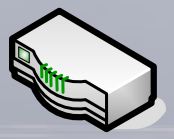
転送テーブル
101 → 4
102 → 5
103 → 2

転送要請はホストID:Aとペアの秘密鍵で署名する



パケット転送エージェント

転送テーブル
101 → 4
102 → 5
103 → 2



ダブルジャンプ問題対策

転送要請はホストID:Aとペアの秘密鍵で署名する

転送

MOVE

ホストID:A

1

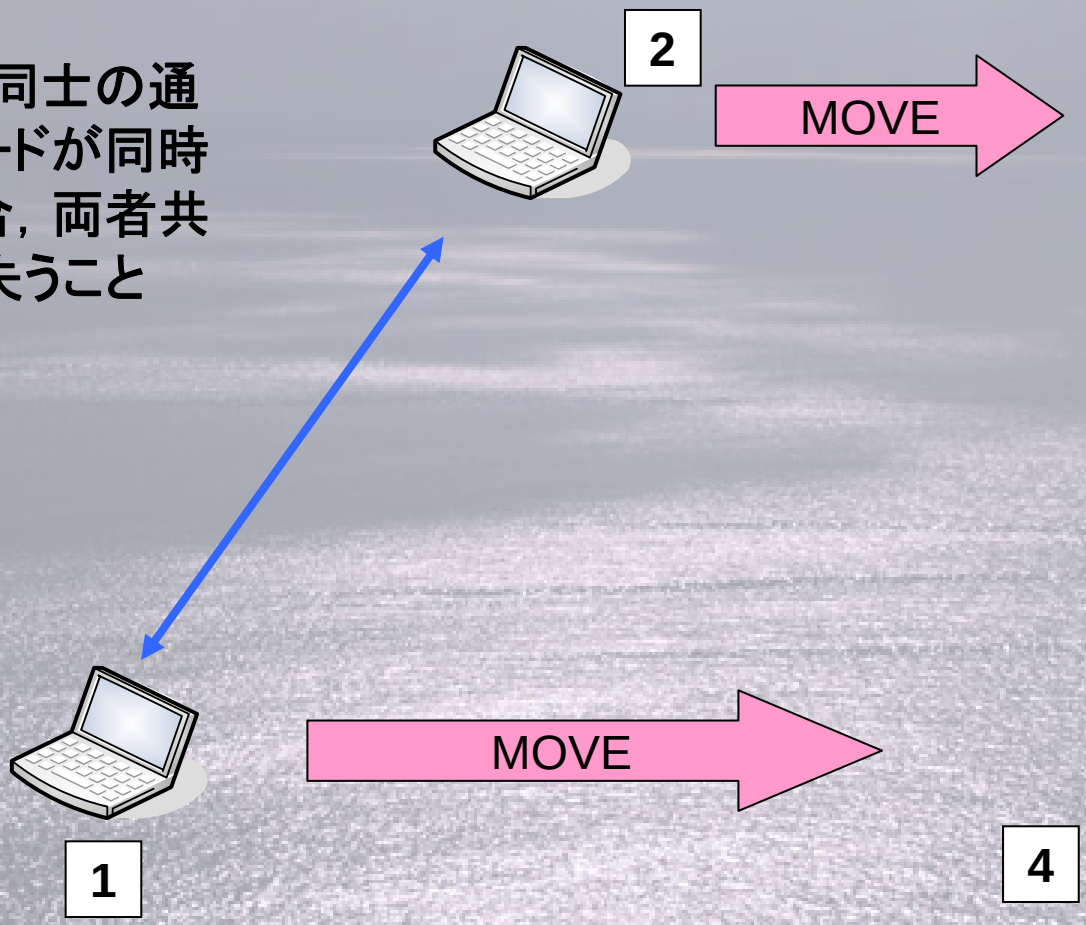
2

ダブルジャンプ問題

モバイルノード同士の通信で両方のノードが同時に移動した場合，両者共に通信先を見失うこと

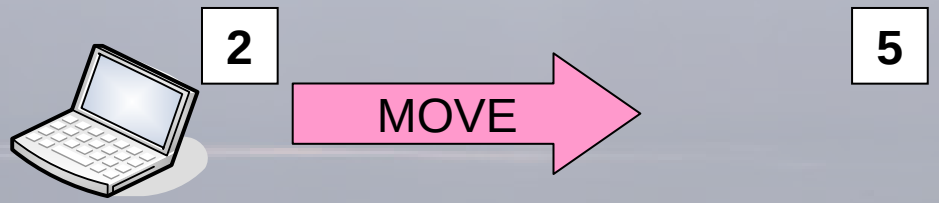
ダブルジャンプ問題

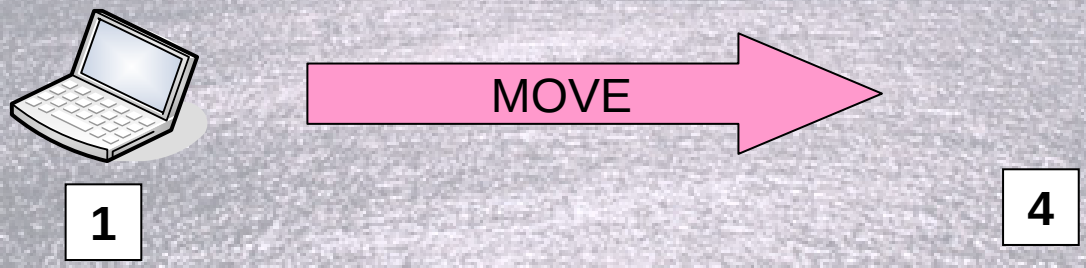
モバイルノード同士の通信で両方のノードが同時に移動した場合，両者共に通信先を見失うこと



ダブルジャンプ問題

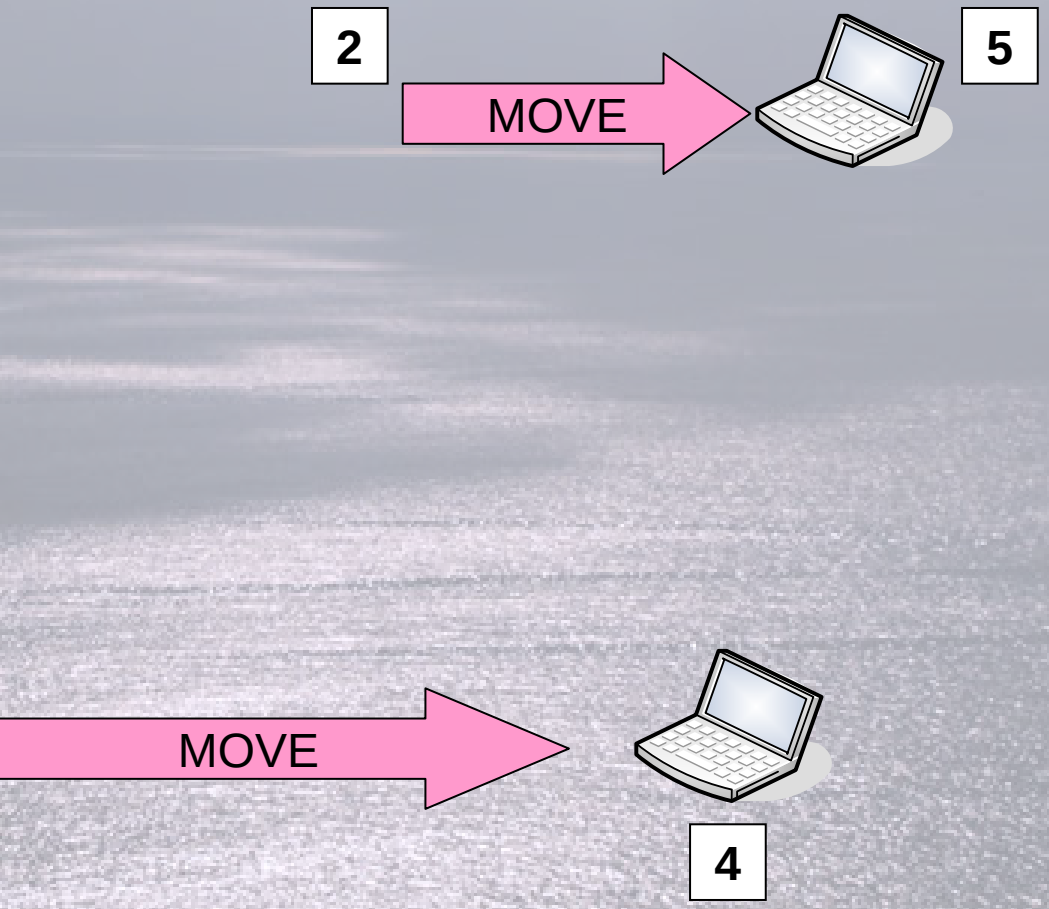
モバイルノード同士の通信で両方のノードが同時に移動した場合，両者共に通信先を見失うこと





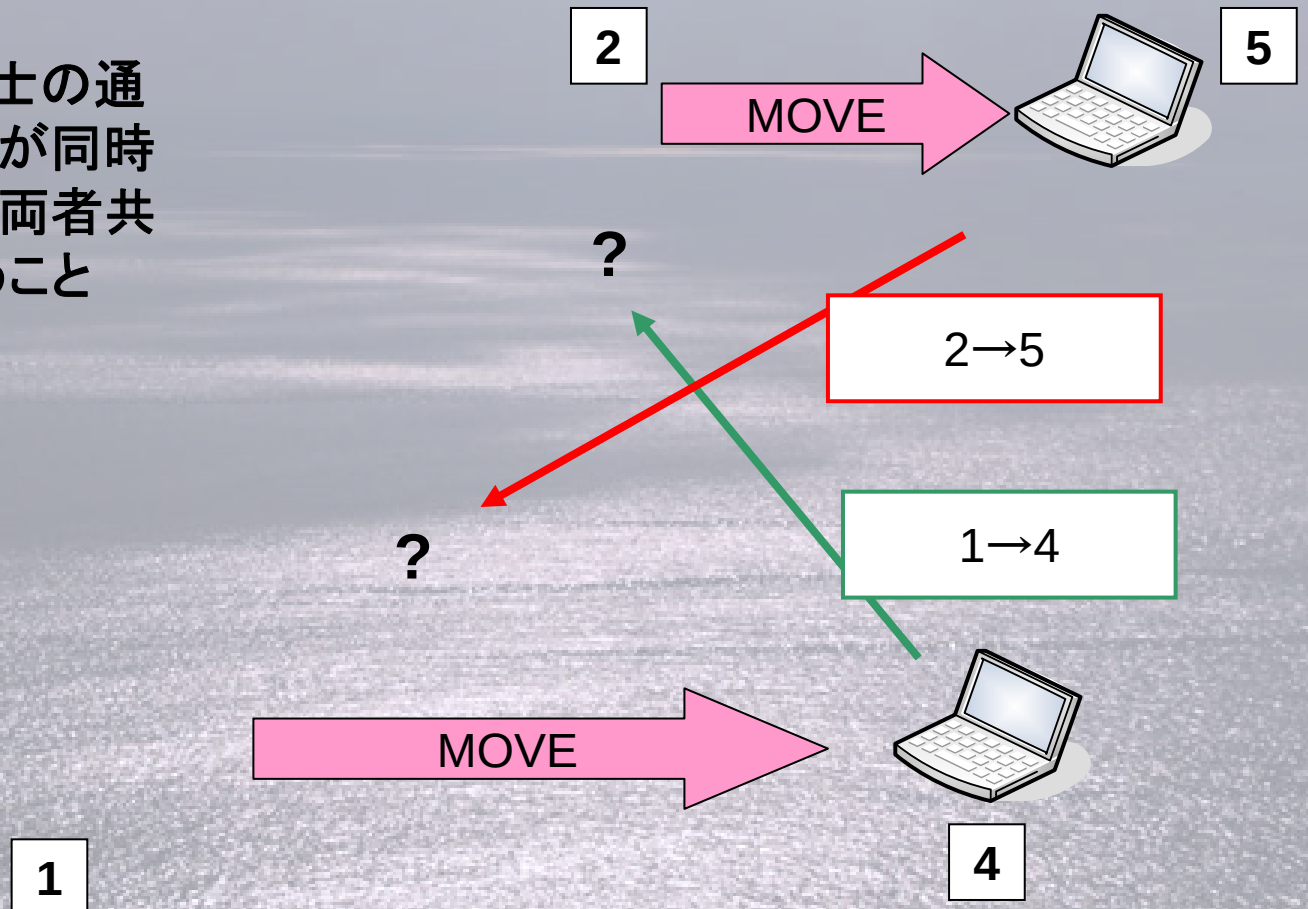
ダブルジャンプ問題

モバイルノード同士の通信で両方のノードが同時に移動した場合，両者共に通信先を見失うこと



ダブルジャンプ問題

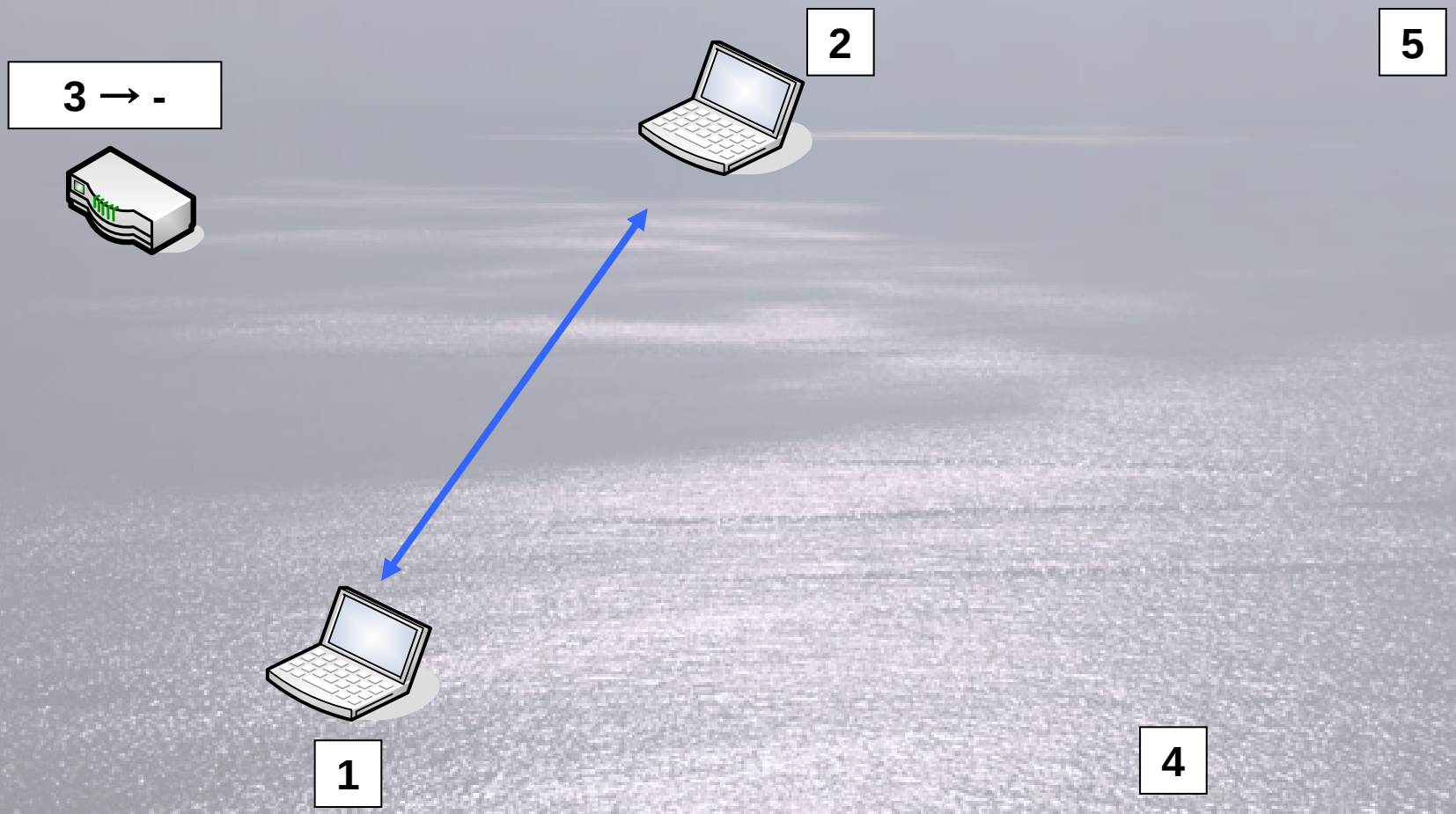
モバイルノード同士の通信で両方のノードが同時に移動した場合，両者共に通信先を見失うこと



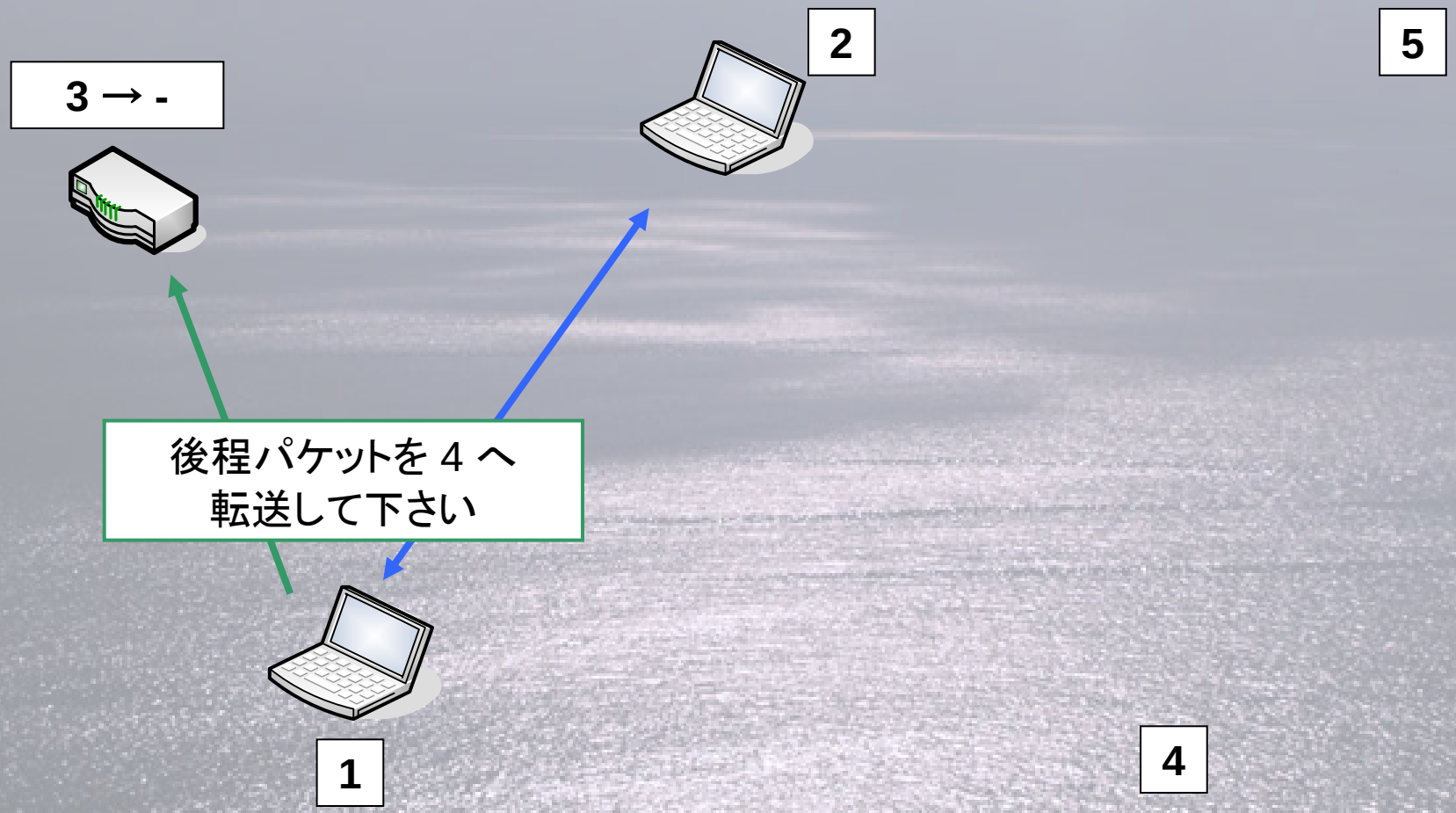
ダブルジャンプ問題対策

- パケット転送エージェントを用いることでこの問題への対策となる(らしいがどのように組み込むのか不明・・・・・・)
- 要するに恐らく・・・

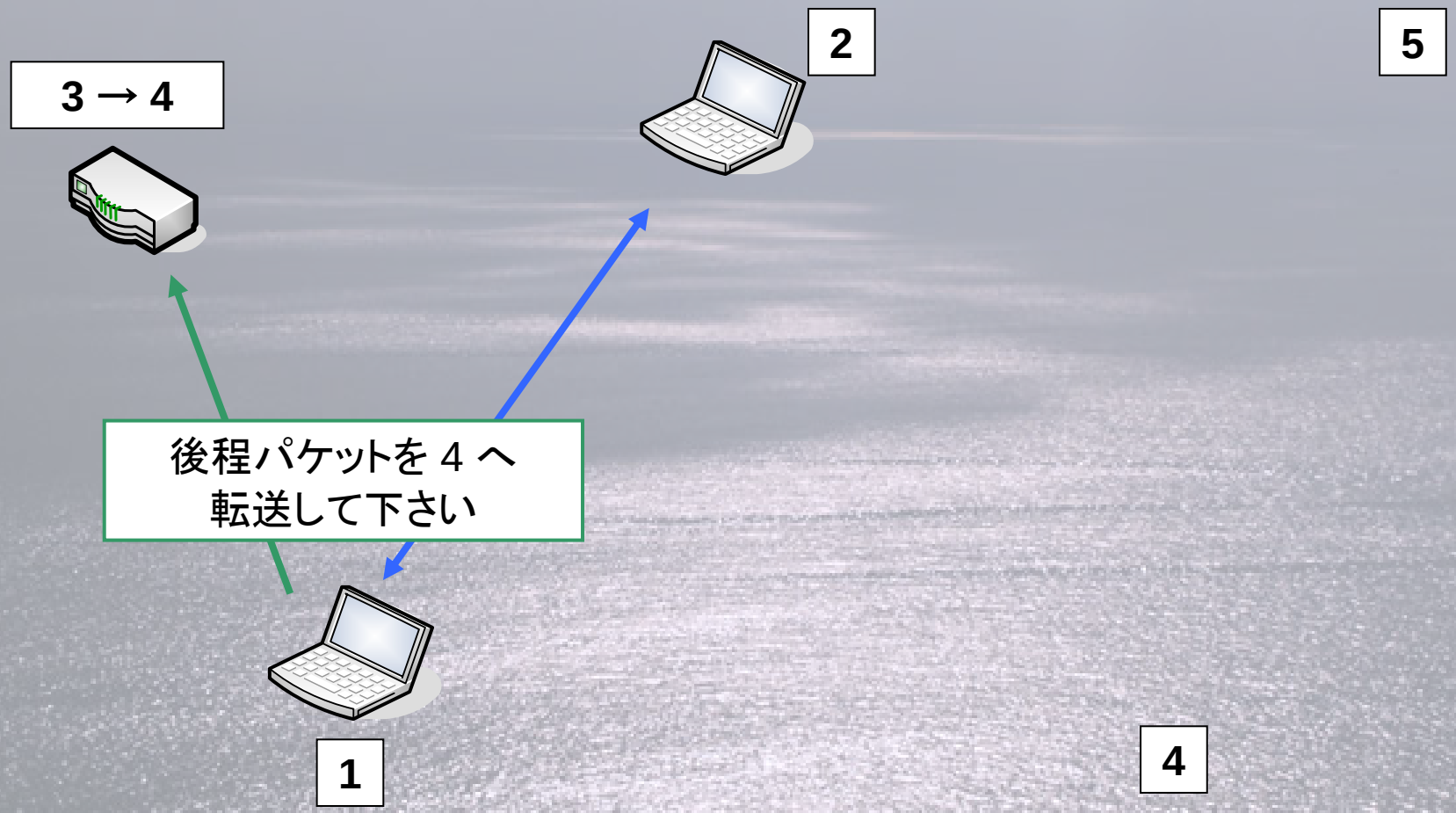
ダブルジャンプ問題対策（妄想）



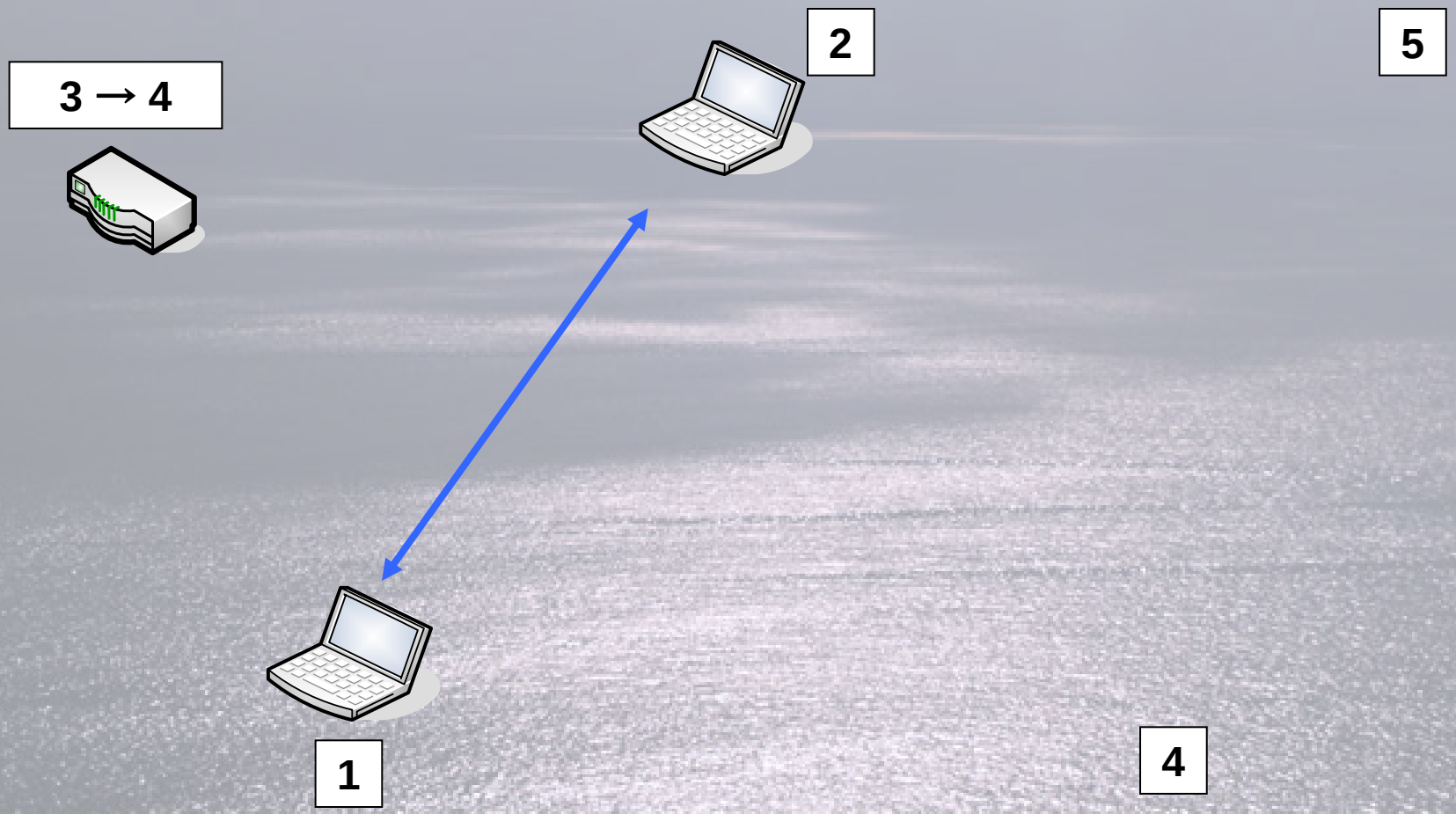
ダブルジャンプ問題対策（妄想）



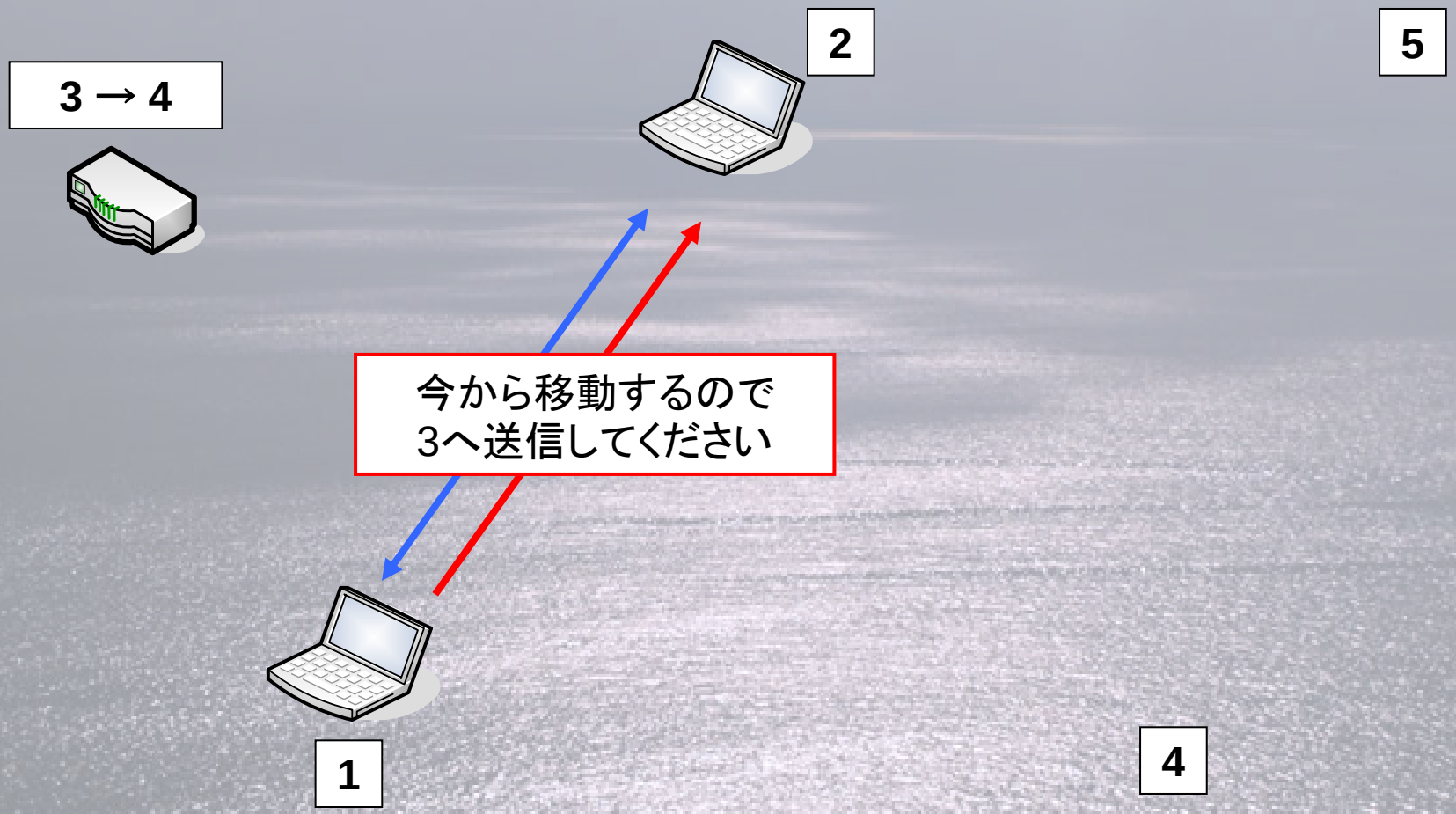
ダブルジャンプ問題対策（妄想）



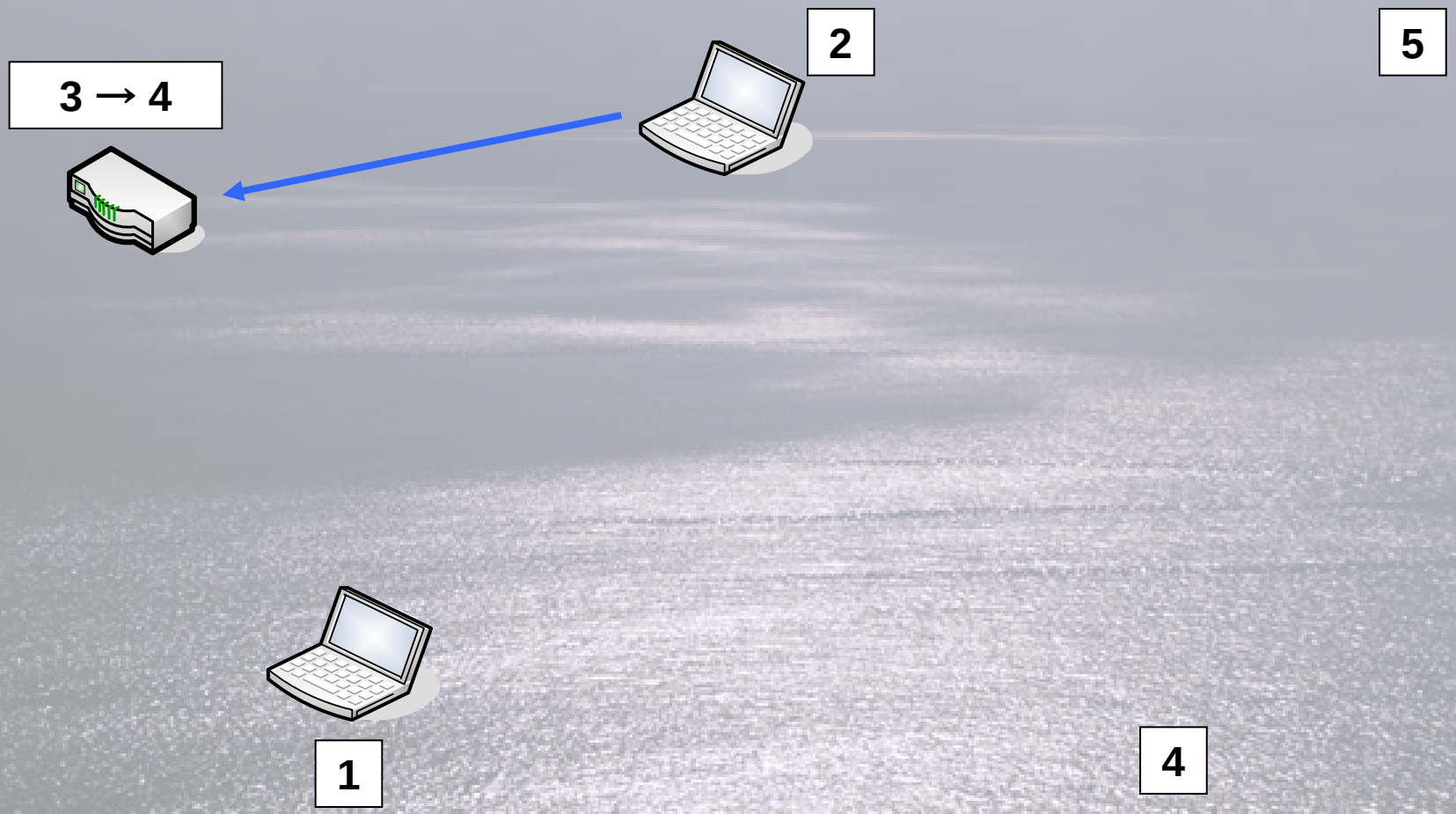
ダブルジャンプ問題対策（妄想）



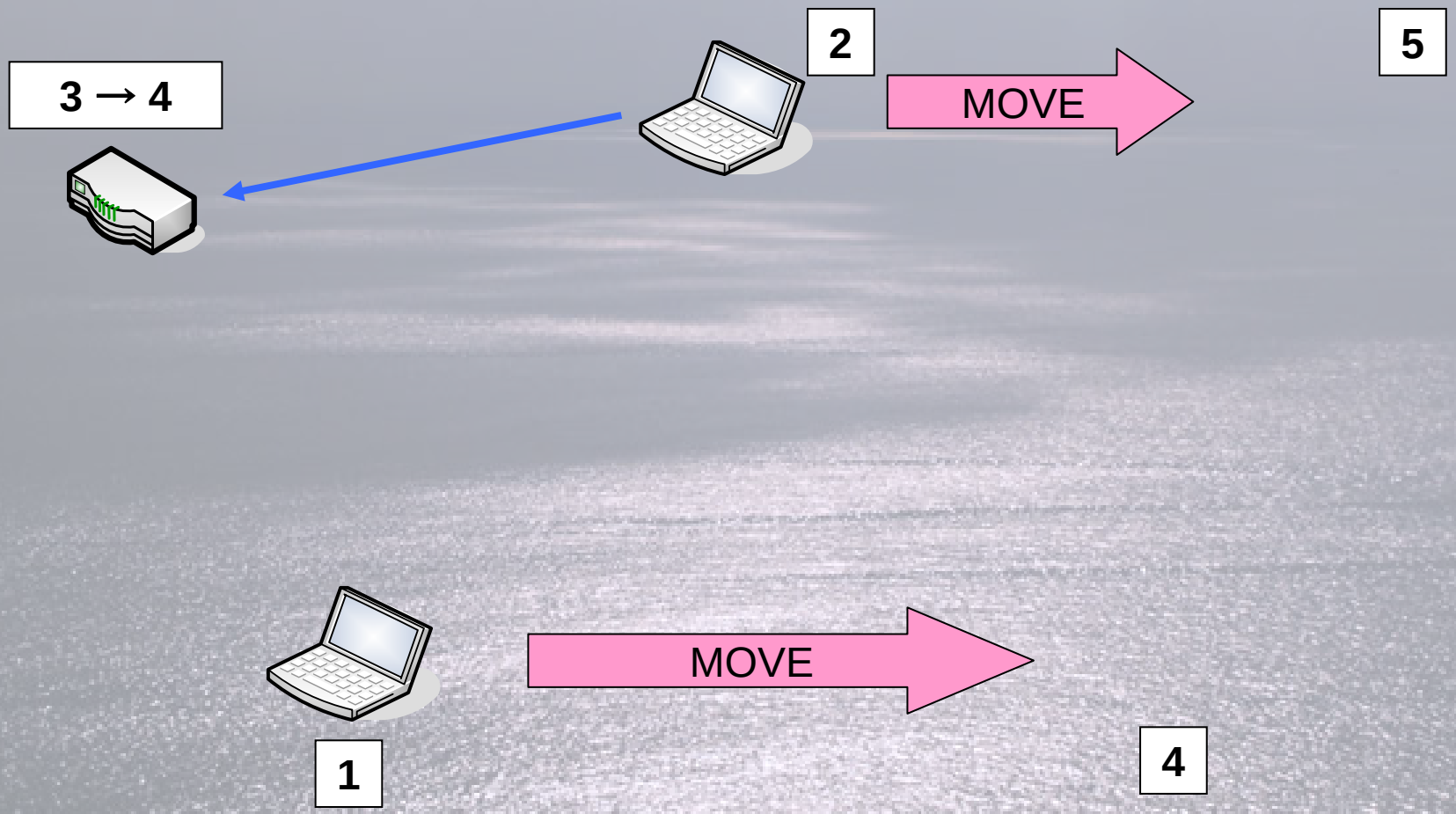
ダブルジャンプ問題対策（妄想）



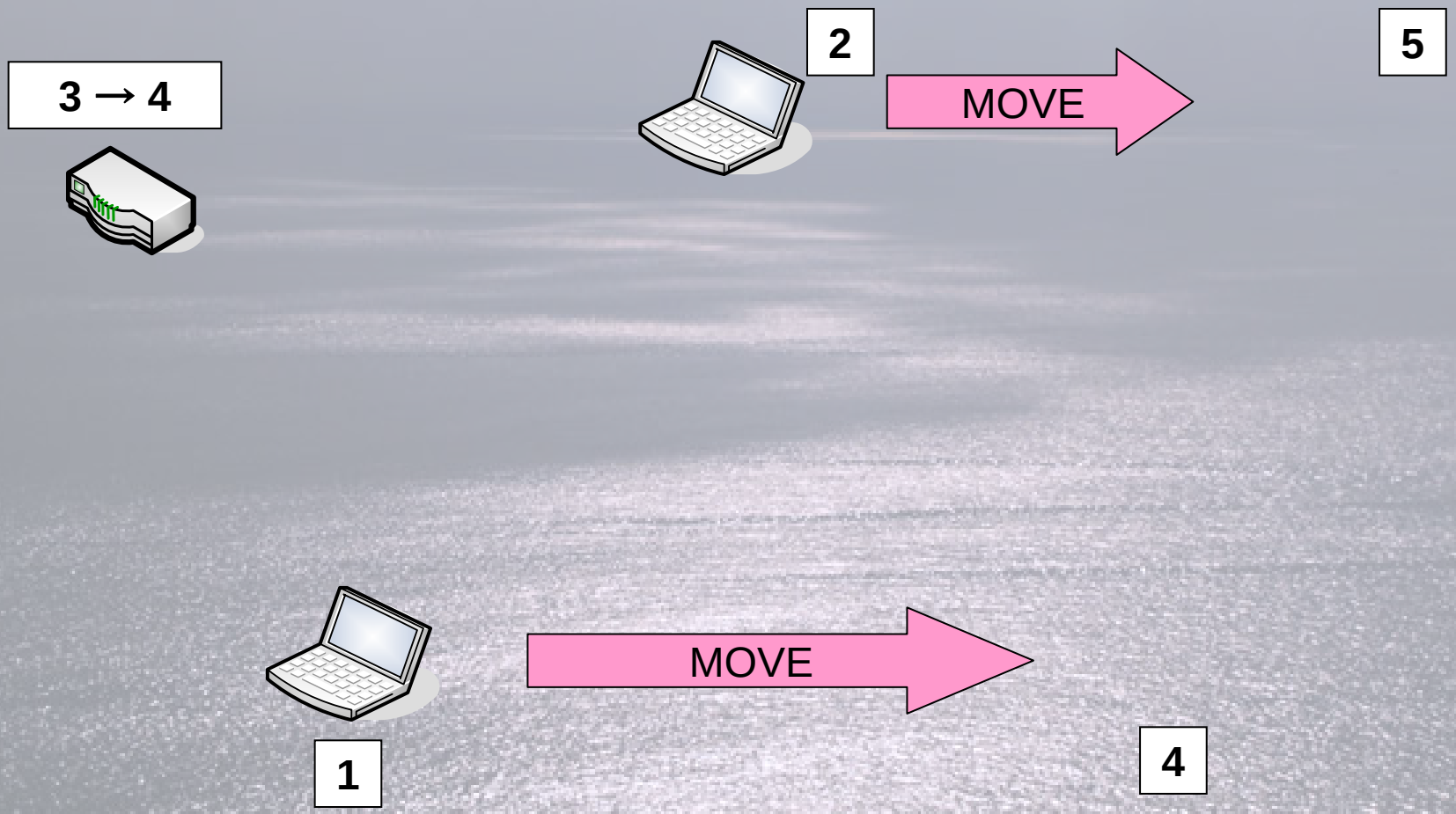
ダブルジャンプ問題対策（妄想）



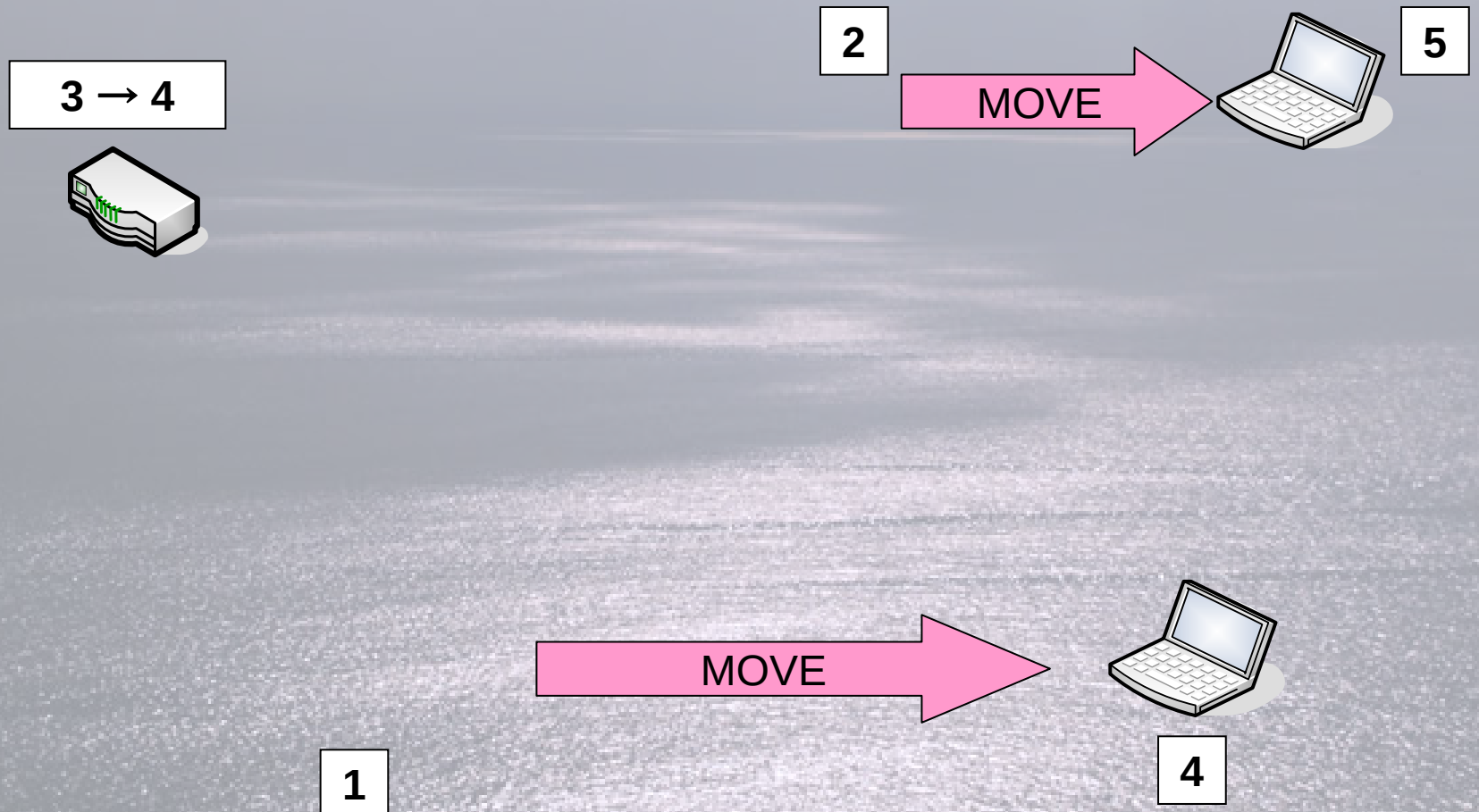
ダブルジャンプ問題対策（妄想）



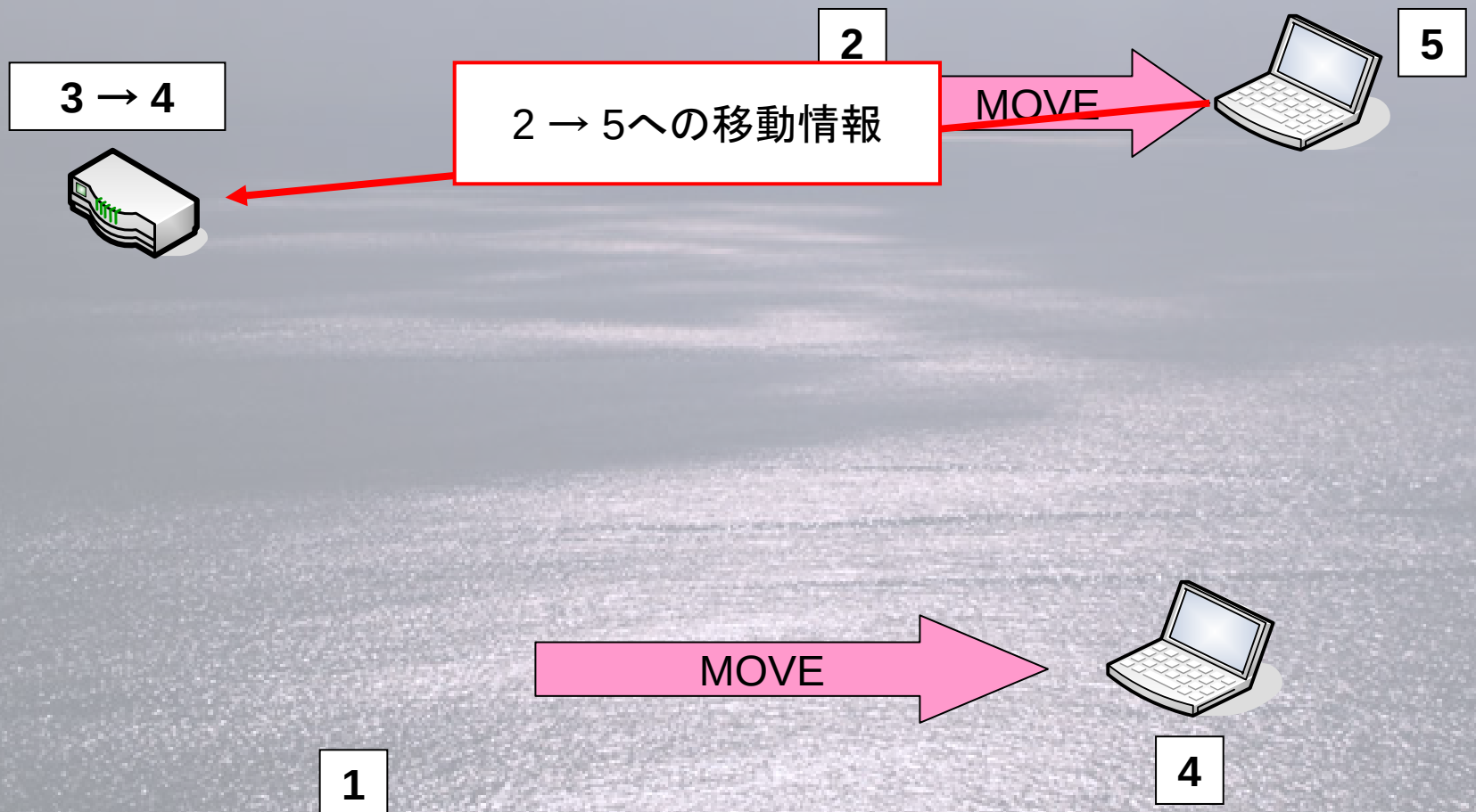
ダブルジャンプ問題対策（妄想）



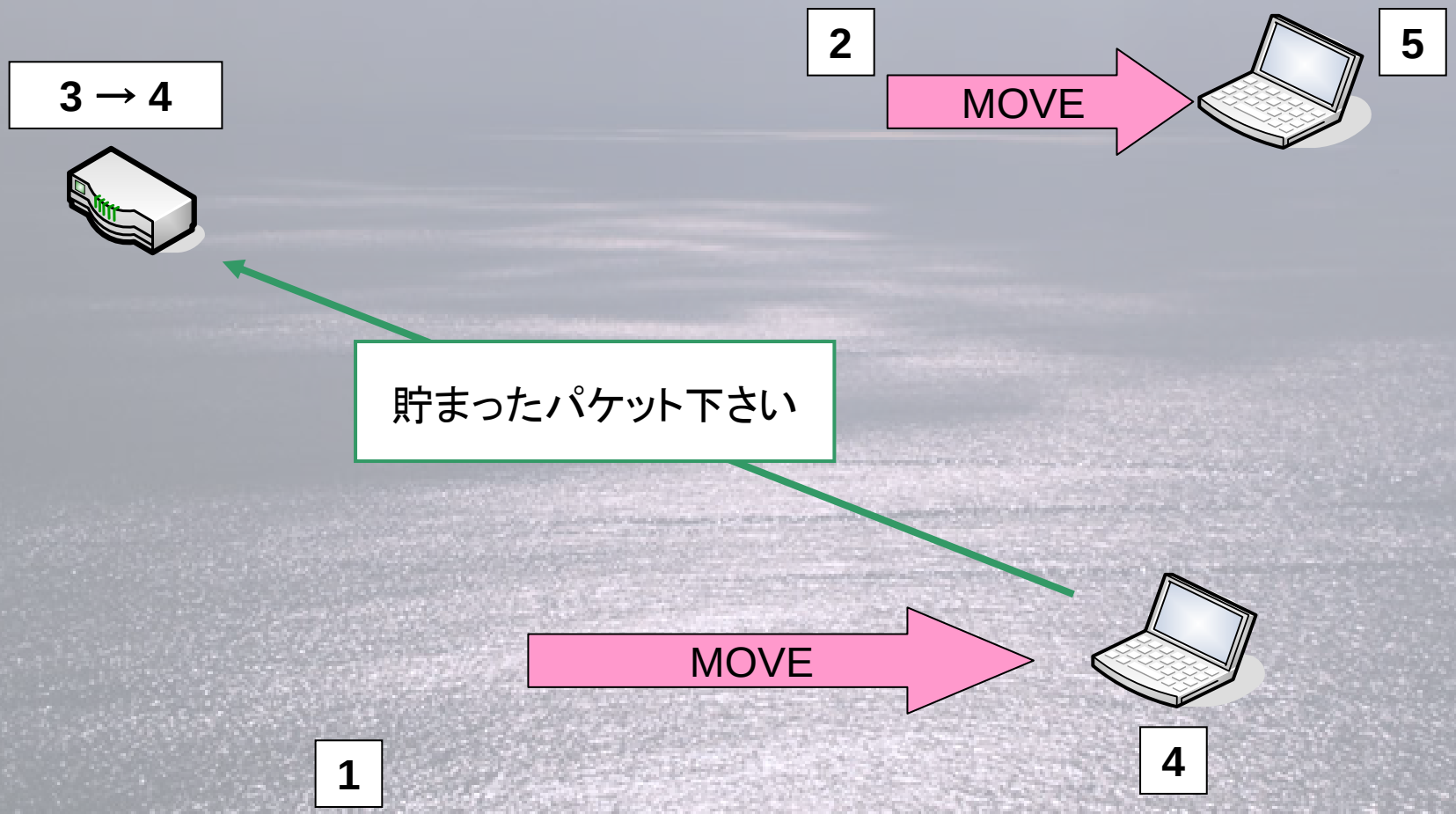
ダブルジャンプ問題対策（妄想）



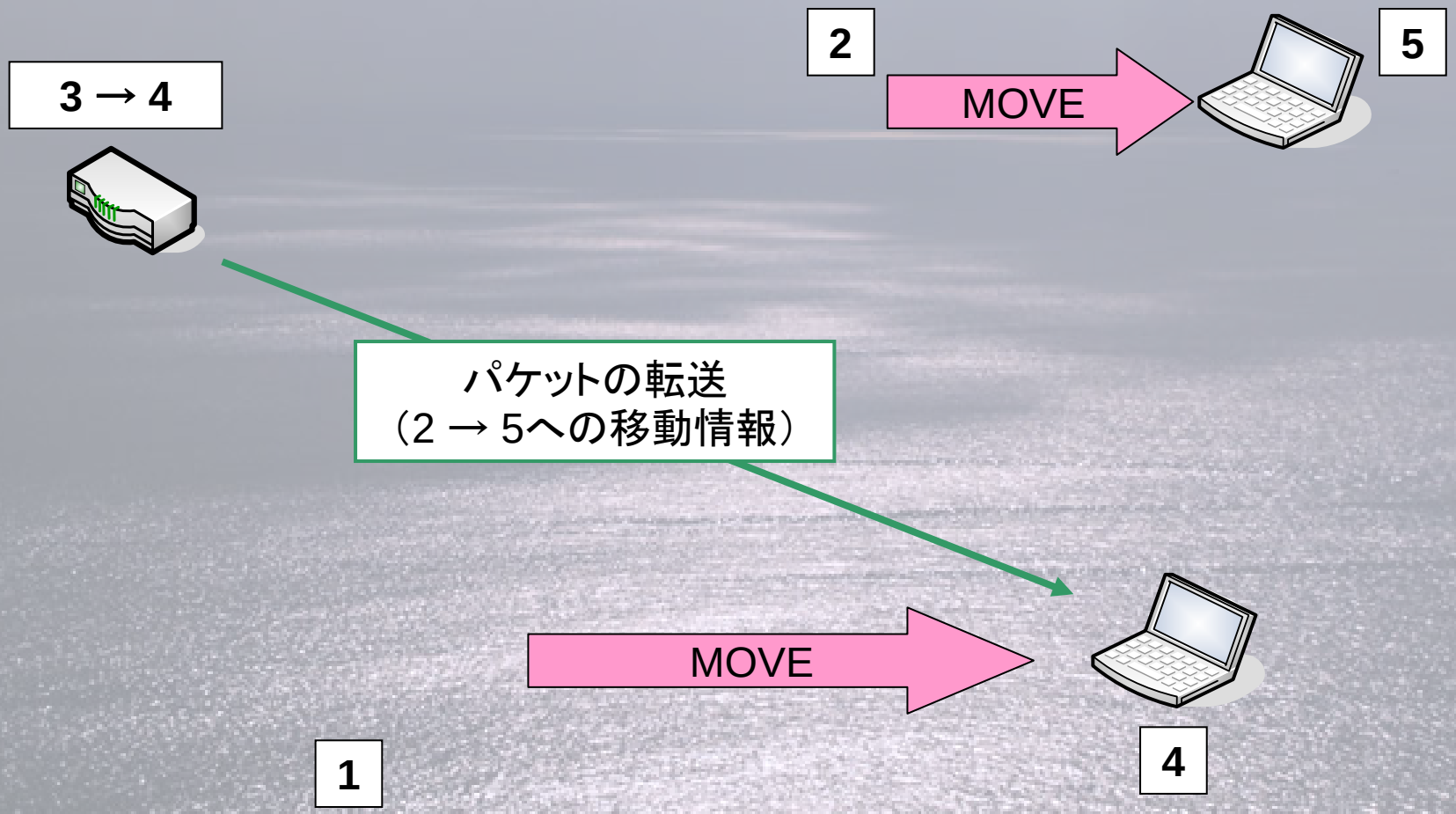
ダブルジャンプ問題対策（妄想）



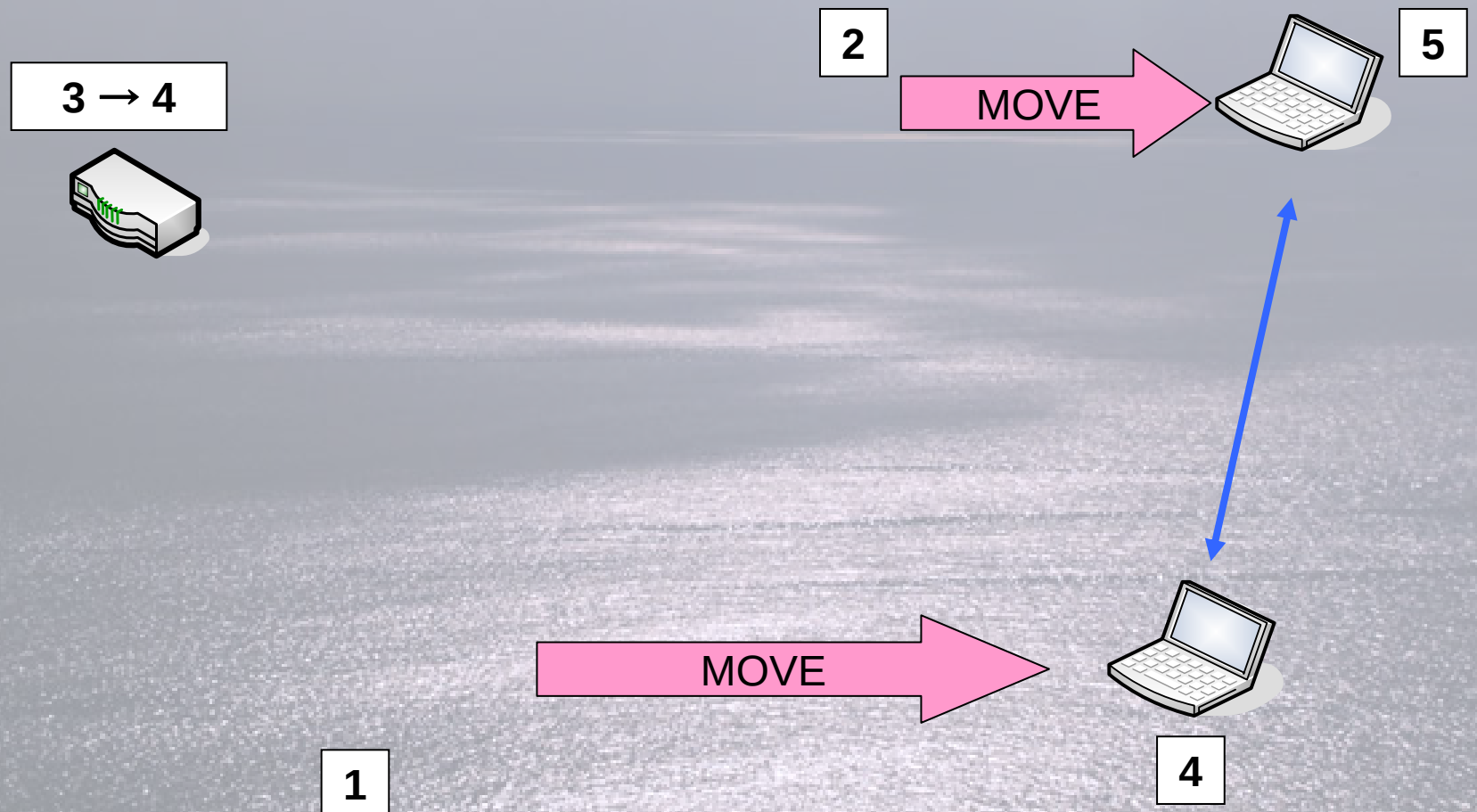
ダブルジャンプ問題対策（妄想）



ダブルジャンプ問題対策（妄想）



ダブルジャンプ問題対策（妄想）



まとめ

- HIPアーキテクチャについて述べた
- 位置情報とホスト情報を分離することによるモビリティとマルチホームの実現について述べた

以上