Designing a Resource Pooling Transport Protocol

Michio Honda, Keio University

Elena Balandina, Nokia Research Center

Pasi Sarolahti, Nokia Research Center

Lars Eggert, Nokia Research Center

Global Internet Symposium

April 24, 2009

Nokia Research Center

NOKIA

Outline

1. Background
2. Motivation
3. Design requirements for current commercial Internet
4. Resource pooling protocol (RPP) overview
5. Path discovery mechanism of RPP
6. Sequence numbers
7. Packet scheduling
8. Congestion control
9. Conclusion and ongoing work

Nokia Research Center

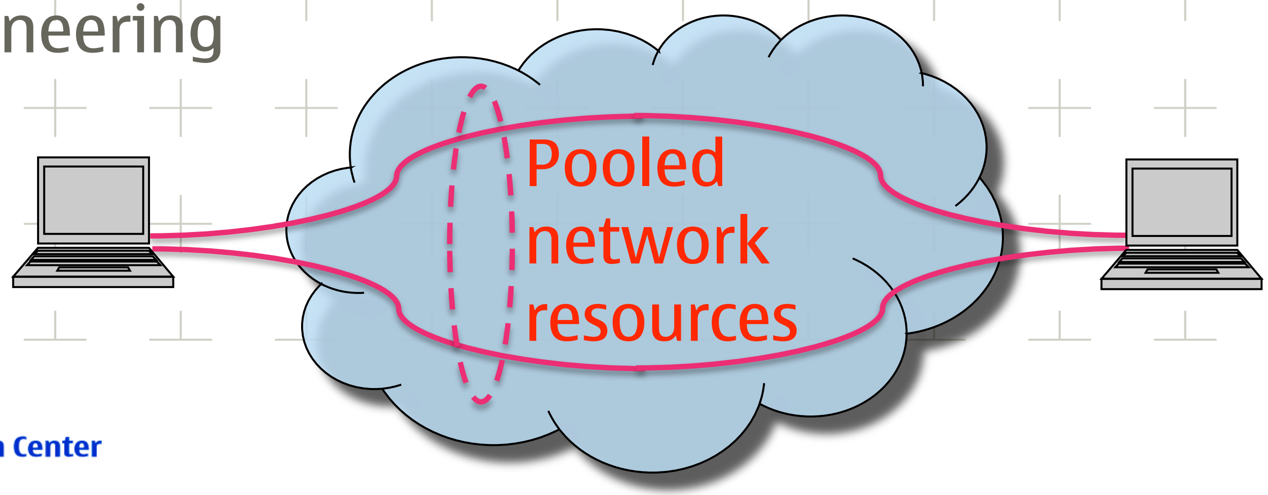
NOKIA

Background

End-to-end communication and routing is usually along a single path

Resource pooling

- Aggregate network path resources as one
- higher resource utilization, better robustness, and traffic engineering



Motivation for Resource Pooling Protocol (RPP)

Simultaneous multipath utilization by end system functionality

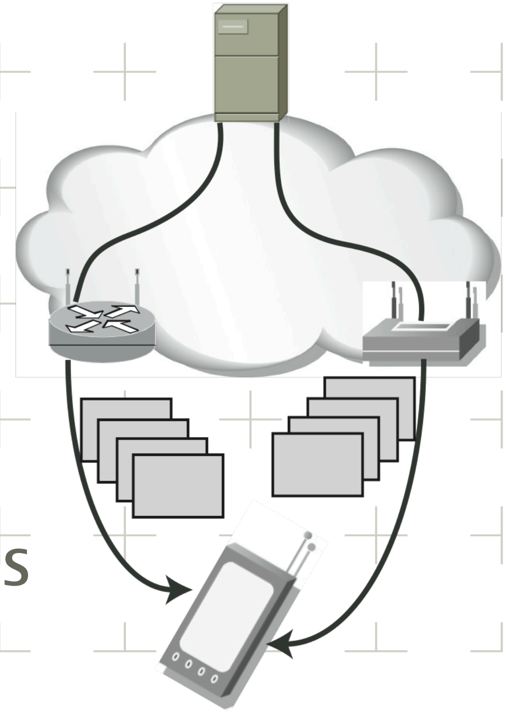
- Many hosts are equipped with multiple network interfaces
- Aggregate spare bandwidth of multiple access links

RPP transmits data over multiple paths

No practical multipath transport protocols exists in terms of ``deployability''

- Middlebox transparency (i.e., NATs, firewalls)

RPP is a new transport protocol that focuses on deployability of multipath transport protocols



Nokia Research Center

NOKIA

Design requirements for current commercial Internet

For deployment, we need

- Standard TCP application transparency
  - We must not enforce modification of existing TCP applications
  - We must prevent protocol backward compatibility with TCP
    - Should be able to communicate with both RPP and TCP peers
- Middlebox transparency
  - Connectivity is frequently restricted by the middleboxes (e.g., NAT)
  - Transport protocols allowed by NAT boxes are only UDP and TCP

TCP extension approach is appropriate

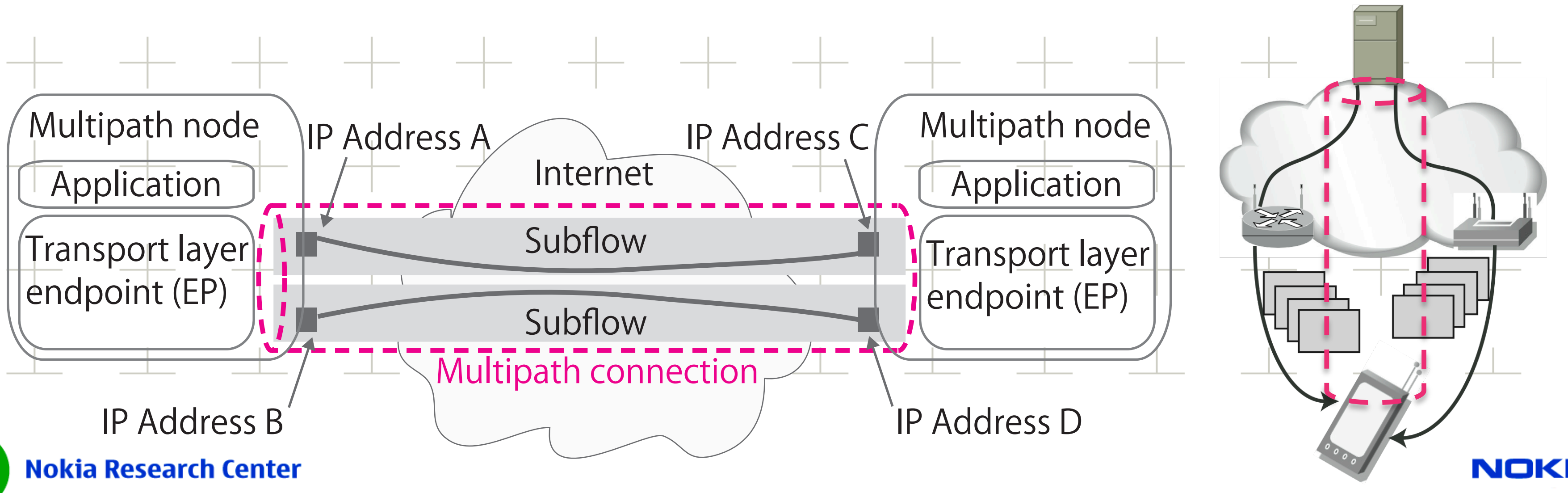
Nokia Research Center NOKIA

RPP overview

Multipath connection

An entity over which applications communicate between transport layer endpoints (EP)

- Provide the same communication primitive as TCP (i.e., a reliable and ordered byte stream) – looks like a TCP connection from the application



Nokia Research Center

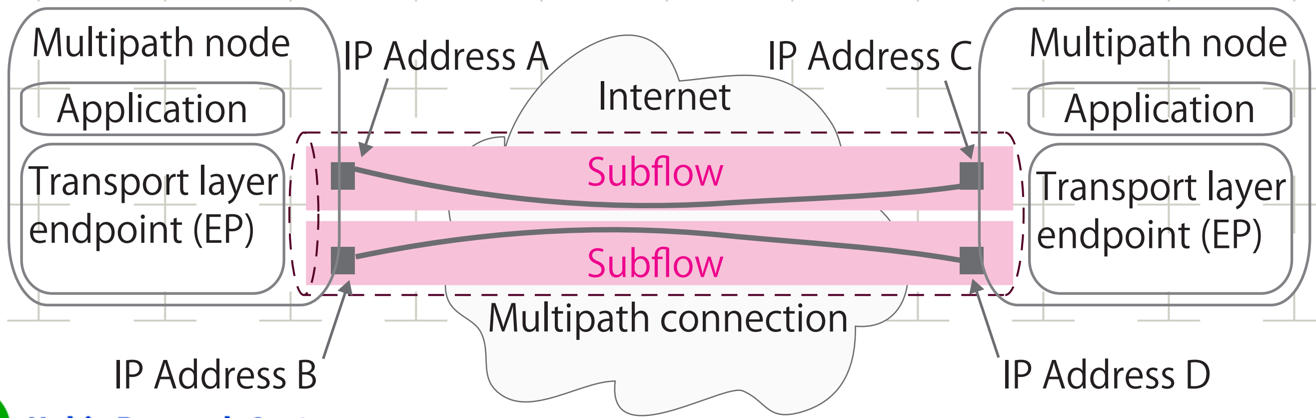
NOKIA

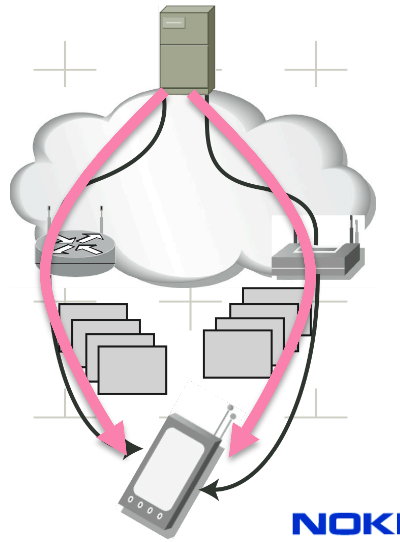
RPP overview

Subflow

An entity over which the endpoint transmits a flow along a path

- look like a TCP connection from the network
- Each chunk of application data contains a TCP header and the TCP protocol number in the IP header
- Established by SYN three-way handshake





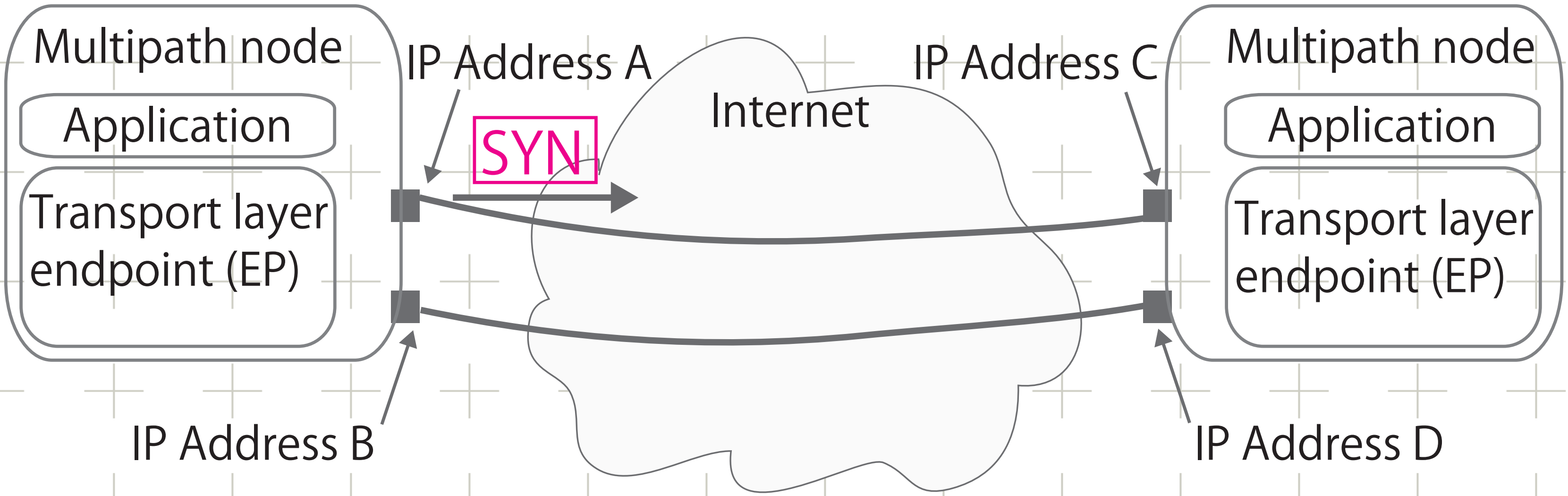
Nokia Research Center

NOKIA

RPP overview - connection / subflow setup

A subflow is initiated by three-way SYN handshake

- This contains an option ``RPP_INIT" that negotiates the RPP capability (if fail, create a standard TCP connection)



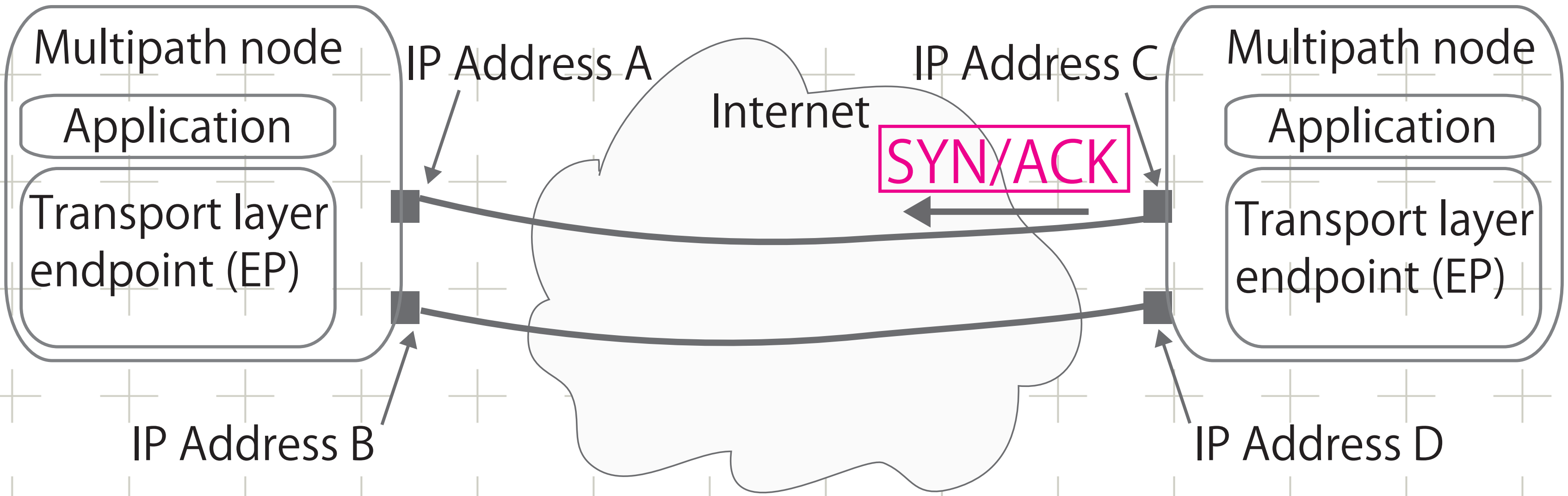
Nokia Research Center

NOKIA

RPP overview - connection / subflow setup

A subflow is initiated by three-way SYN handshake

- This contains an option ``RPP_INIT” that negotiates the RPP capability (if fail, create a standard TCP connection)



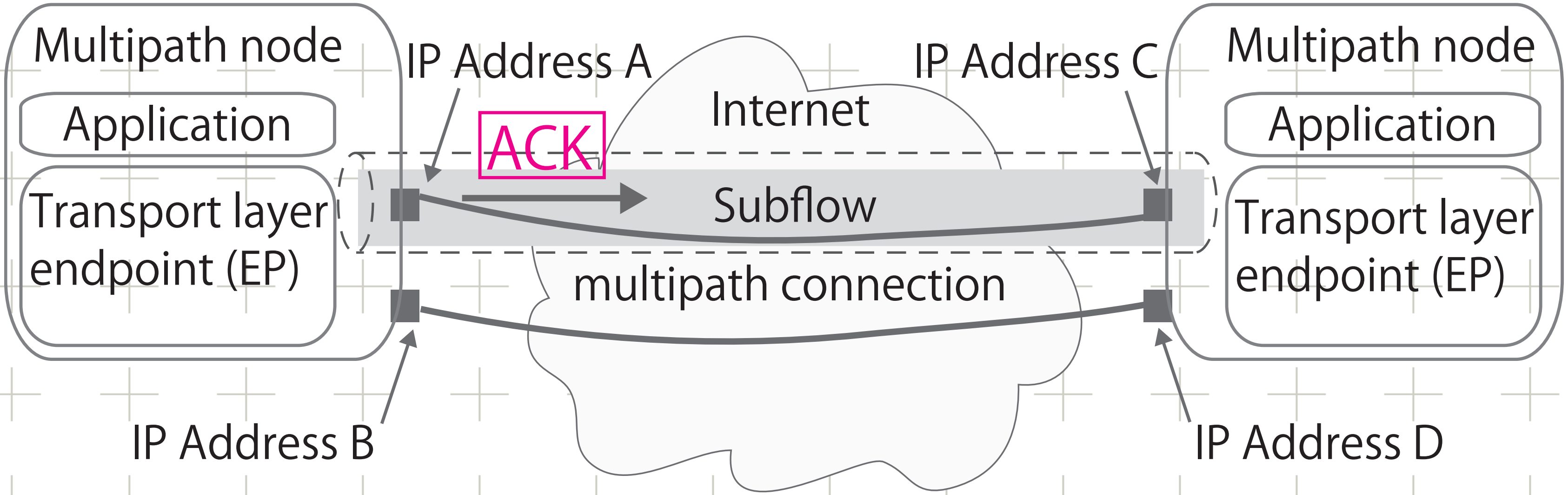
Nokia Research Center

NOKIA

RPP overview - connection / subflow setup

A subflow is initiated by three-way SYN handshake

- This contains an option ``RPP_INIT" that <u>negotiates the RPP capability</u> (if fail, create a standard TCP connection)



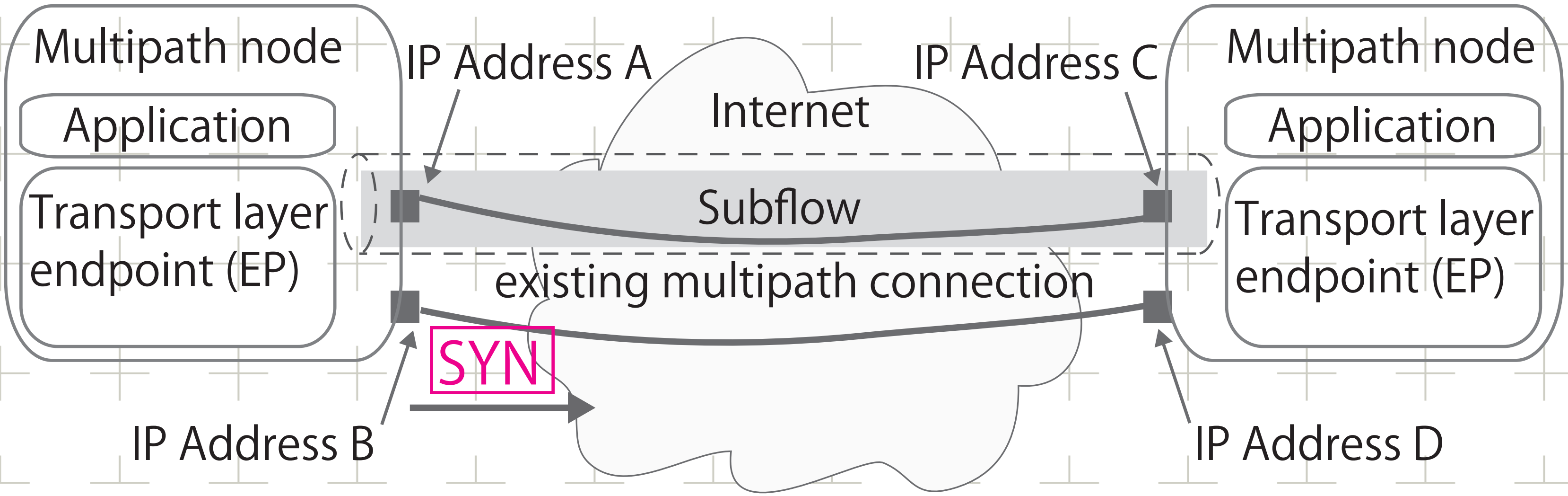
Nokia Research Center

NOKIA

RPP overview - connection / subflow setup

Additional subflow establishment needs to <u>specify the existing multipath connection</u>

Say ``This SYN is for the new subflow joining to the existing connection''



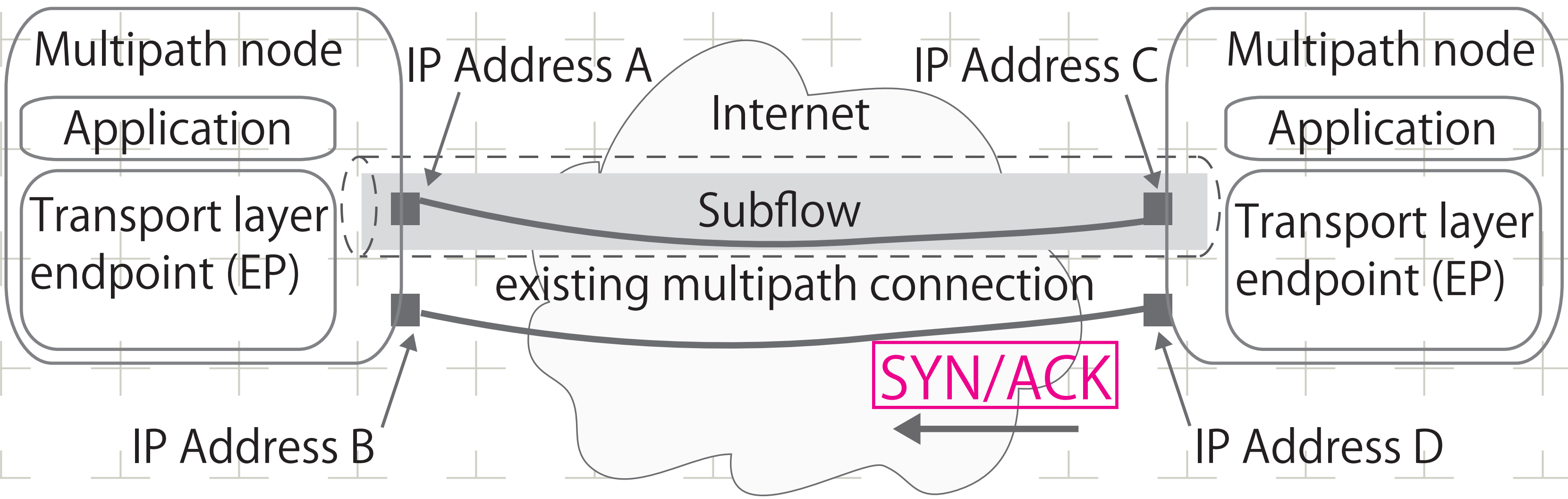
Nokia Research Center

NOKIA

RPP overview - connection / subflow setup

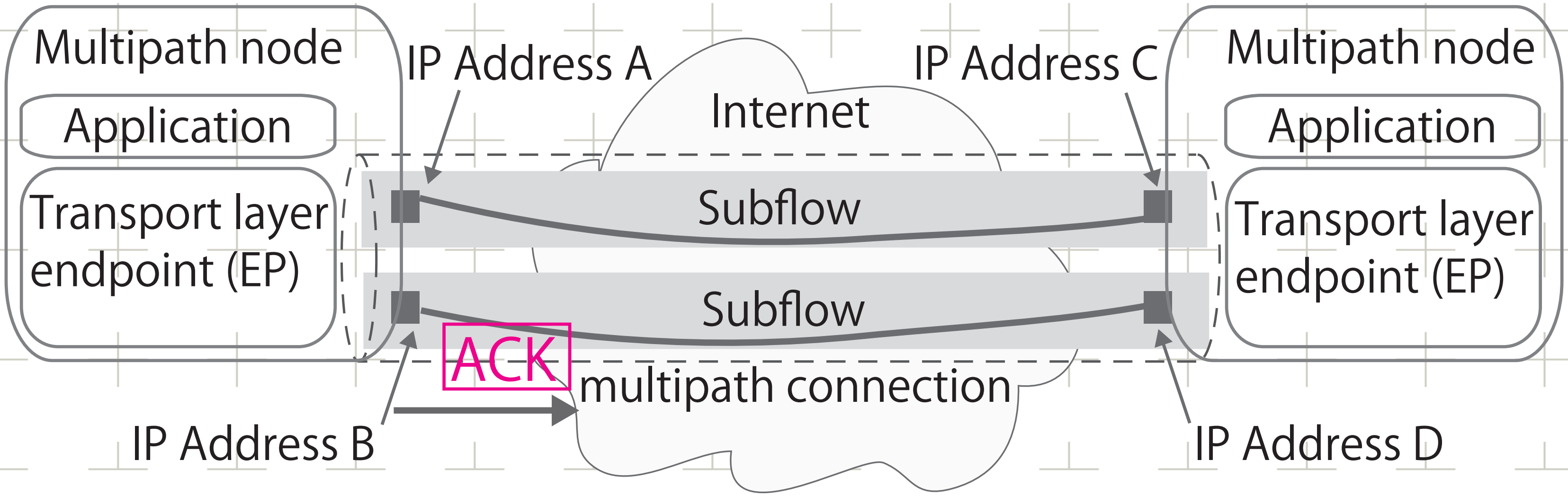
Additional subflow establishment needs to specify the existing multipath connection

Say ``This SYN is for the new subflow joining to the existing connection”



RPP overview - connection / subflow setup

- Additional subflow establishment needs to specify the existing multipath connection
  - Say ``This SYN is for the new subflow joining to the existing connection”



RPP overview - connection identification

How should we describe the existing connection or the first subflow?

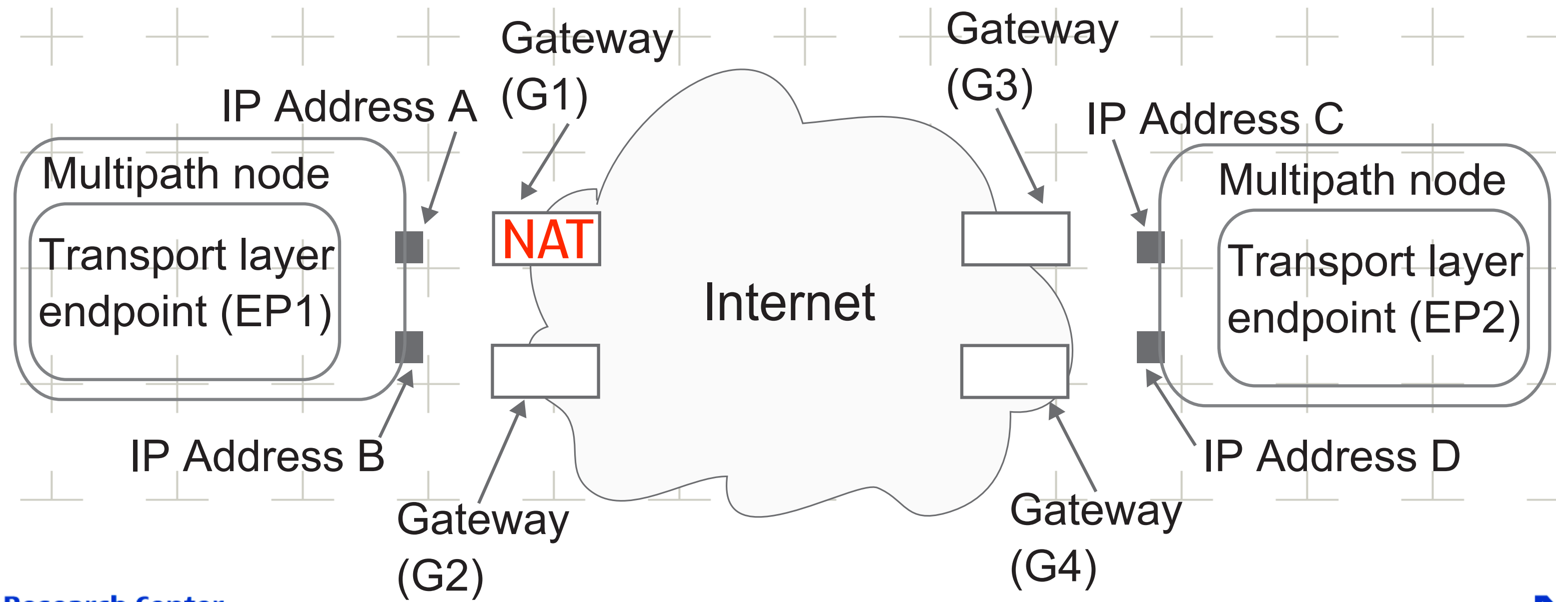
- Source-destination port and address pair might be rewritten at the NATs
- Initial Sequence Number (adopted in AMS) might be rewritten by some firewalls (e.g., pf)
- A new TCP option is appropriate (adopted in pTCP)
  - Dropping probability is 0.3%
- <u>Exchange ``connection ID" at the first subflow setup</u>
  - New subflow SYN specifies the connection ID

Path discovery with subflow establishment

Path discovery should be done by the actual subflow initiation

- not only endpoints but also middle boxes create the state of the subflow

- example: EP1 has looked up EP2's address: "IP Address C"



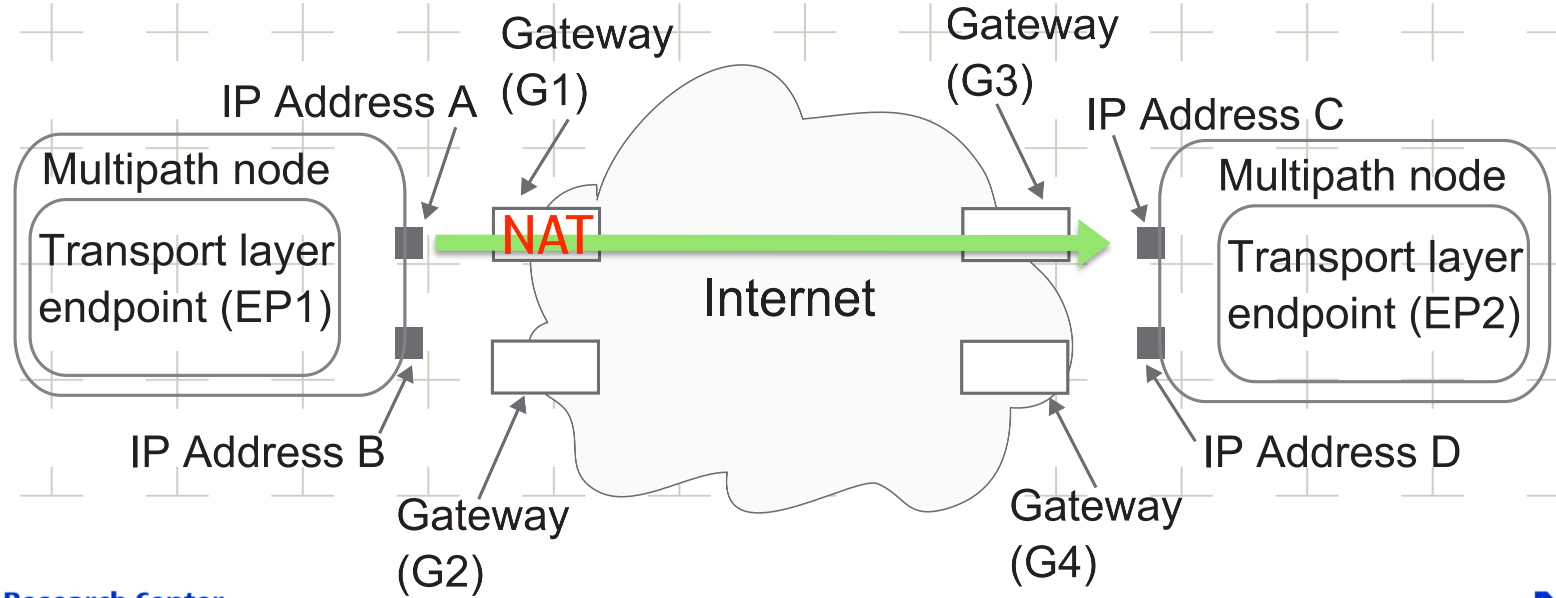
Nokia Research Center

NOKIA

Path discovery with subflow establishment

Path discovery should be done by the actual subflow initiation

- not only endpoints but also middle boxes create the state of the subflow
- EP1 initiates the subflow(A->C)



Nokia Research Center

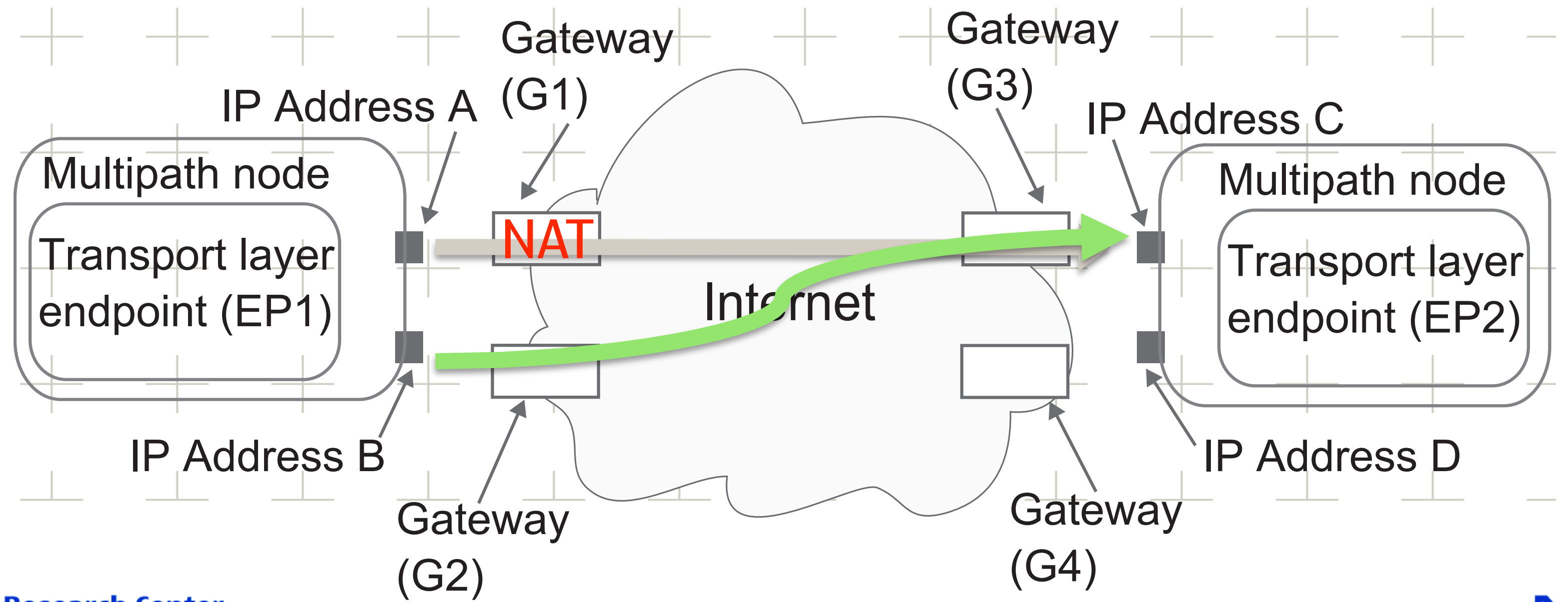
NOKIA

Path discovery with subflow establishment

Path discovery should be done by the actual subflow initiation

- not only endpoints but also middle boxes create the state of the subflow

- EP1 initiates another subflow from another interface

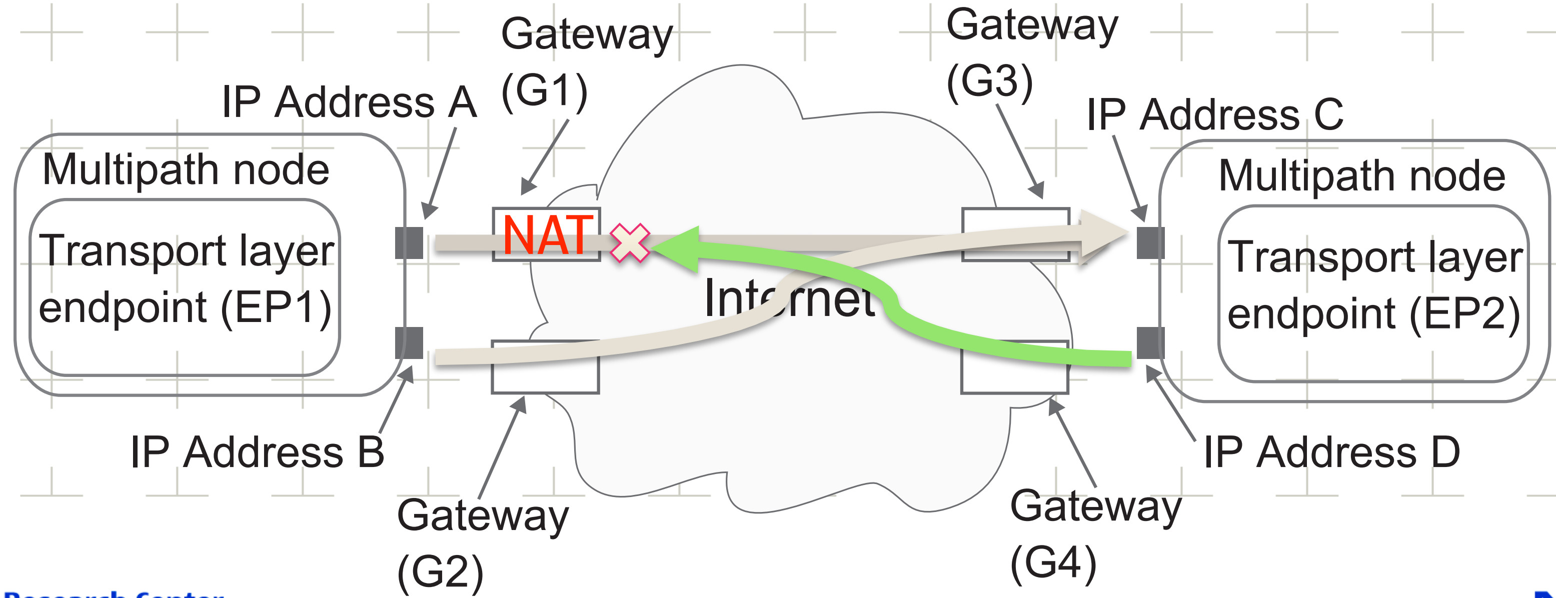


Path discovery with subflow establishment

Path discovery should be done by the actual subflow initiation

- not only endpoints but also middle boxes create the state of the subflow

- EP2 also has another address, but cannot initiate subflow(D->A)



Nokia Research Center

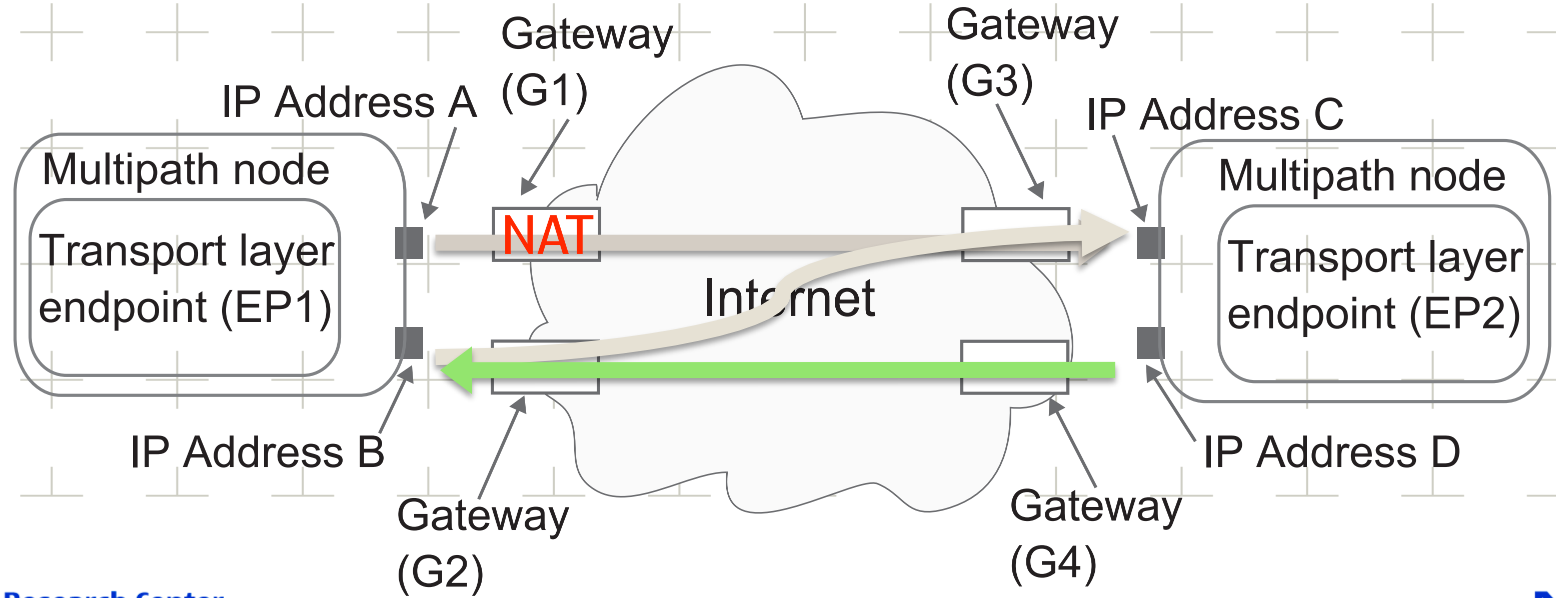
NOKIA

Path discovery with subflow establishment

Path discovery should be done by the actual subflow initiation

- not only endpoints but also middle boxes create the state of the subflow

- EP2 initiates subflow(D->B)



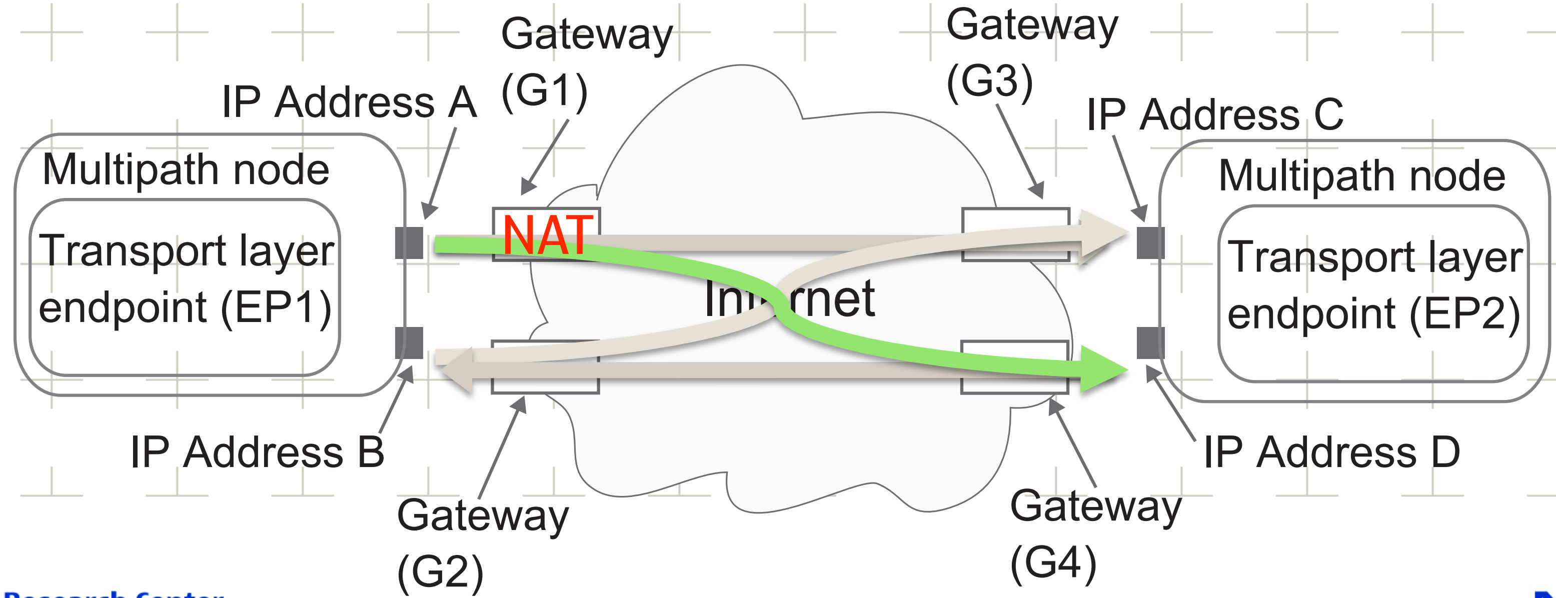
Nokia Research Center

NOKIA

Path discovery with subflow establishment

Path discovery should be done by the actual subflow initiation

- not only endpoints but also middle boxes create the state of the subflow
- EP2 initiates subflow(D->B)



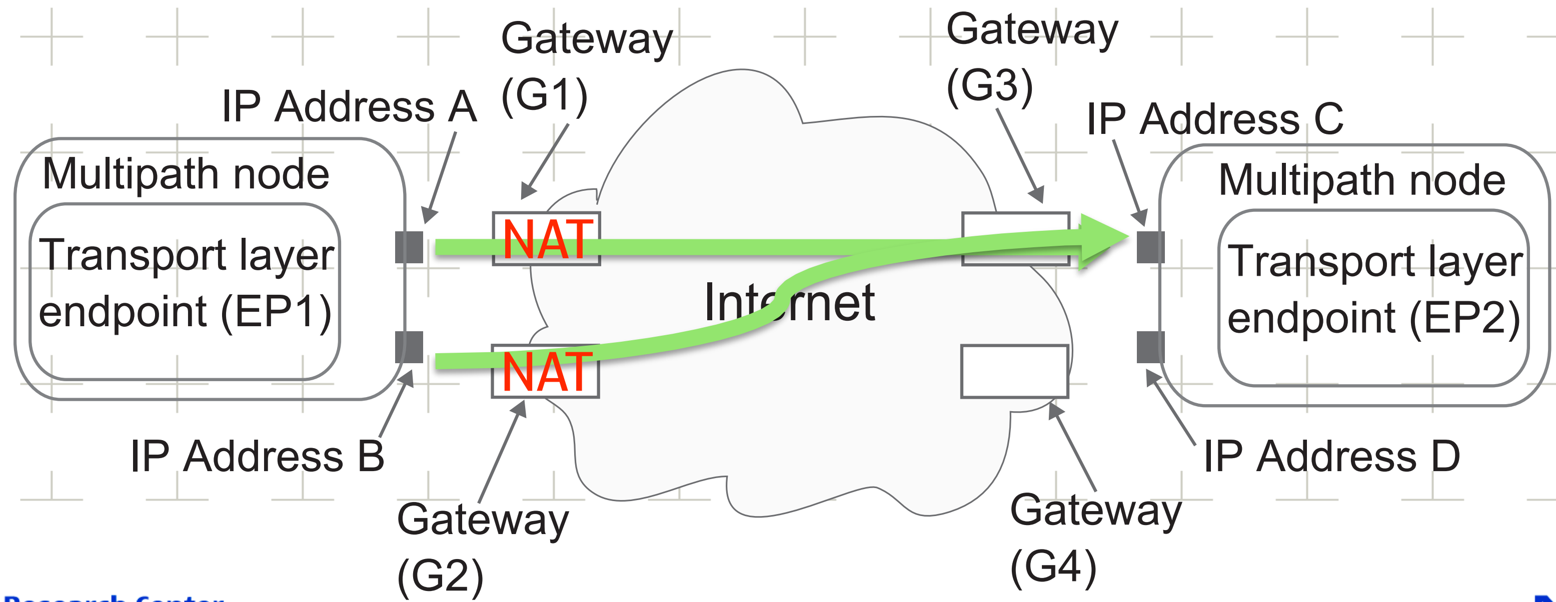
Nokia Research Center

NOKIA

Path discovery with subflow establishment

Explicit notification of another address is required, when either endpoint has only restricted addresses

- EP1 can initiate subflow(A->C) and (B->C)



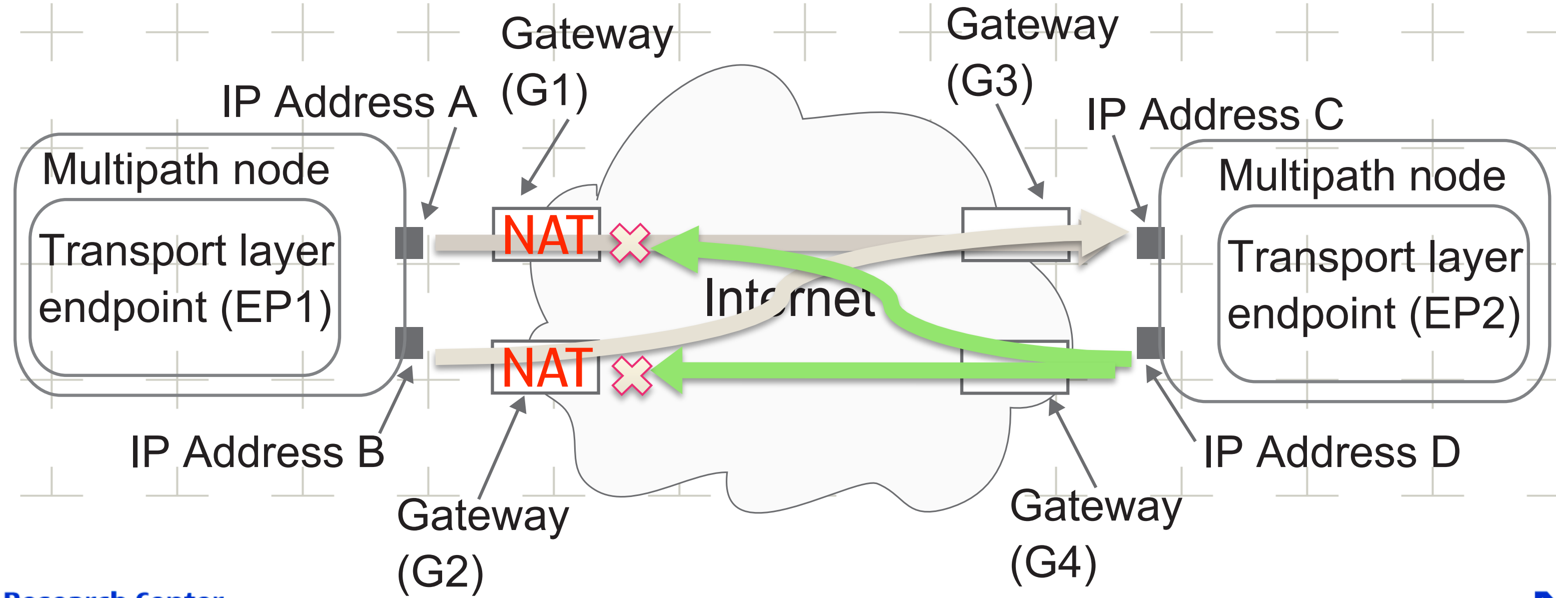
Nokia Research Center

NOKIA

Path discovery with subflow establishment

Explicit notification of another address is required, when either endpoint has only restricted addresses

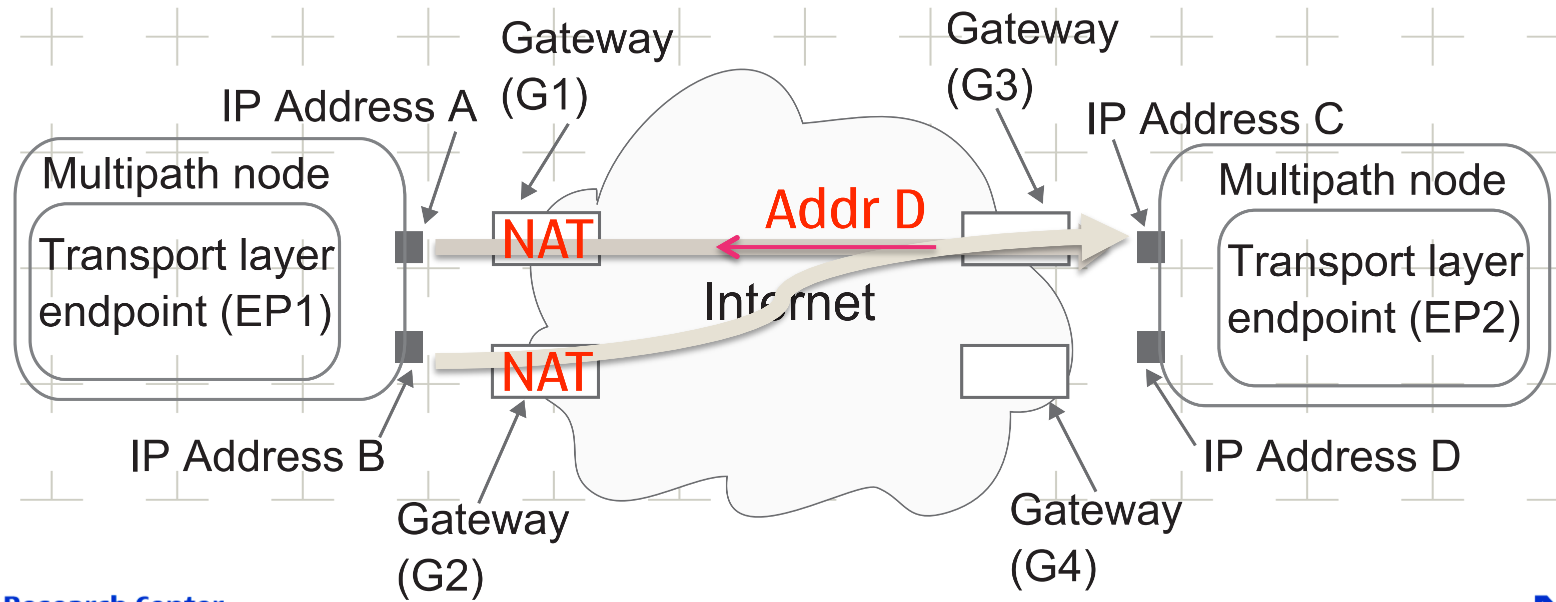
- EP2 cannot initiate subflows from Address D
  - EP1 cannot know existence of Address D



Path discovery with subflow establishment

Explicit notification of another address is required, when either endpoint has only restricted addresses

- EP2 cannot initiate subflows from Address D
  - EP1 cannot know existence of Address D



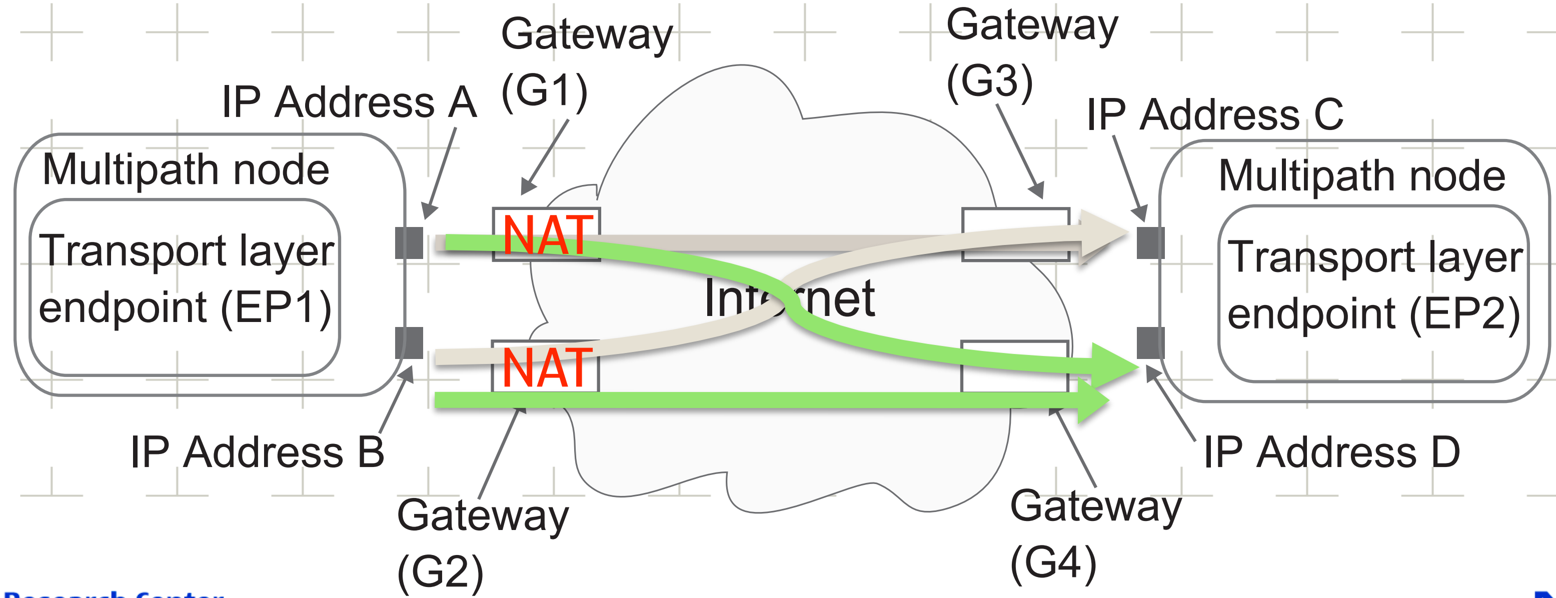
Nokia Research Center

NOKIA

Path discovery with subflow establishment

Explicit notification of another address is required, when either endpoint has only restricted addresses

- EP2 cannot initiate subflows from Address D
  - EP1 cannot know existence of Address D



Nokia Research Center

NOKIA

Sequence Numbers of RPP

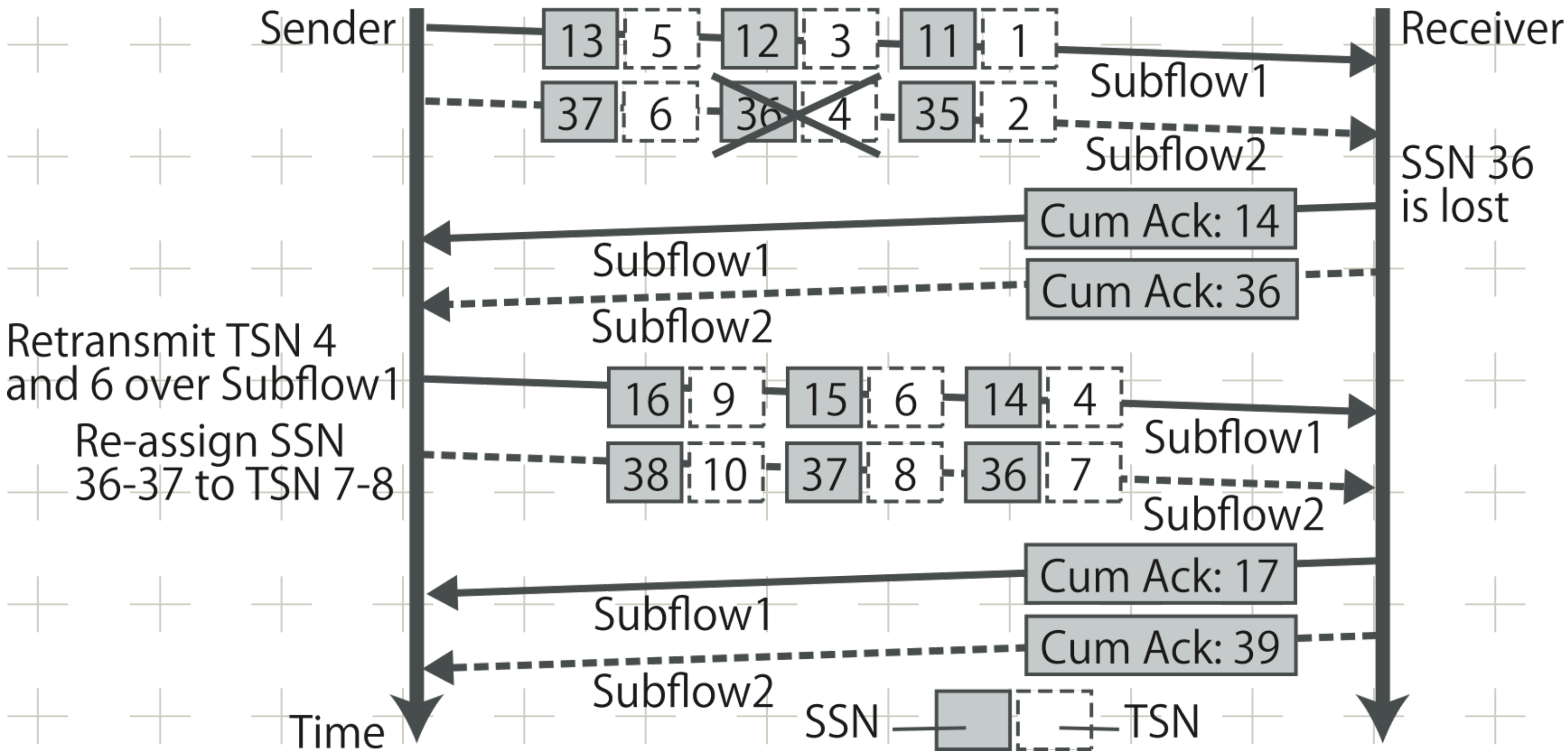
Both subflows and connections require sequence numbers

- Connection-level sequence number (TSN)
  - application-data order
- Subflow-level sequence number (SSN)
  - loss detection on path-basis packet loss

Acknowledgments should occur per subflow

- If we use only connection-level sequence numbers, SACK is enforced so that the sender detects the path which the packets are lost
- Score board could be very large, due to frequent out-of-ordered arriving
- Implementing path congestion control based on loss-feedback could be very complicated

Sequence Numbers of RPP



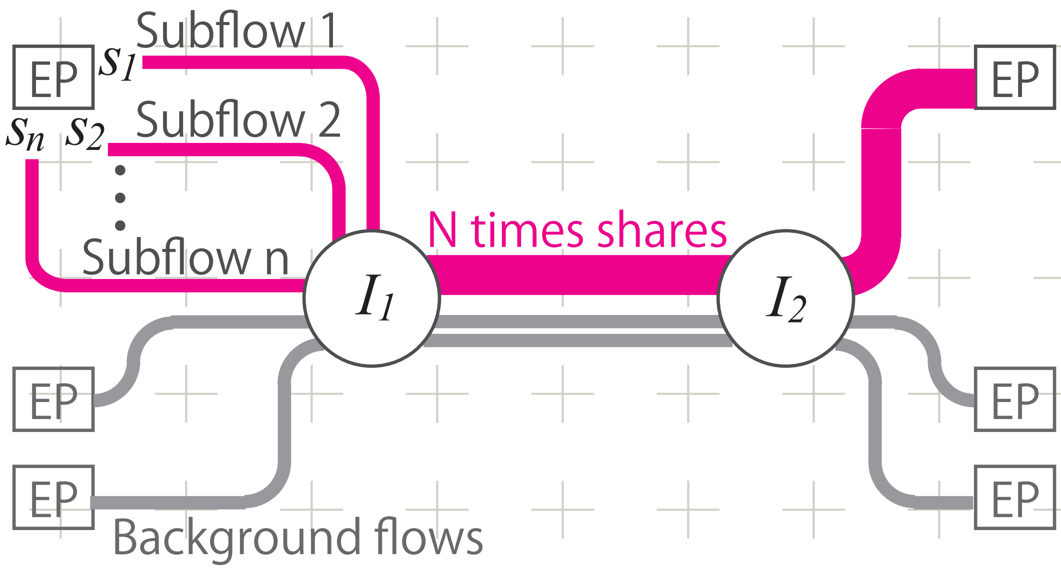
- Connection-level sequence number (TSN)
- Subflow-level sequence number (SSN)

Packet scheduling

Packet scheduling is important to deliver receiving data to application as soon as possible

- propagation delay could be different between subflow
- Out-of-order arriving packets also increase usage of the buffer space at both sender and receiver endpoints
- We are now investigating the algorithm, but several existing papers have also investigated it

Congestion control



Congestion control is basically done per subflow, with the same algorithm as TCP as like existing papers

- It could be overly aggressive at the shared bottleneck in the TCP-friendliness sense
- This problem is being investigated in another paper

Conclusion and progress

A new transport protocol that uses resources along multiple paths as ``a pool of resources"

- Deployable design principle
  - A multipath connection looks like a TCP connection from the application
  - A subflow looks like a TCP connection from the network
- Implementation is work in progress in Linux 2.6 kernel

Current limitation

How do we deal with proxy nodes?

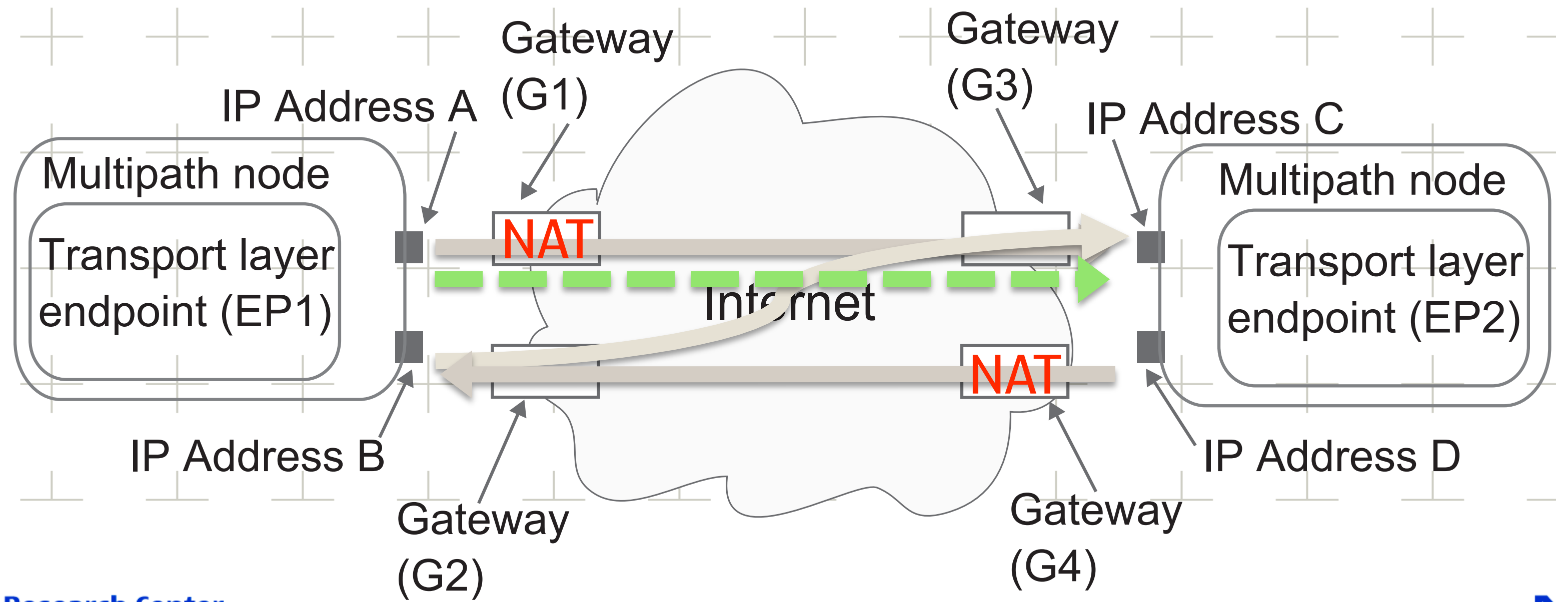
they might not forward TCP options to another connection

but even if proxy nodes exist, standard TCP connection can be established

Path discovery with subflow establishment

Subflows are also available between NAT boxes

- EP1 sends a SYN from A to C



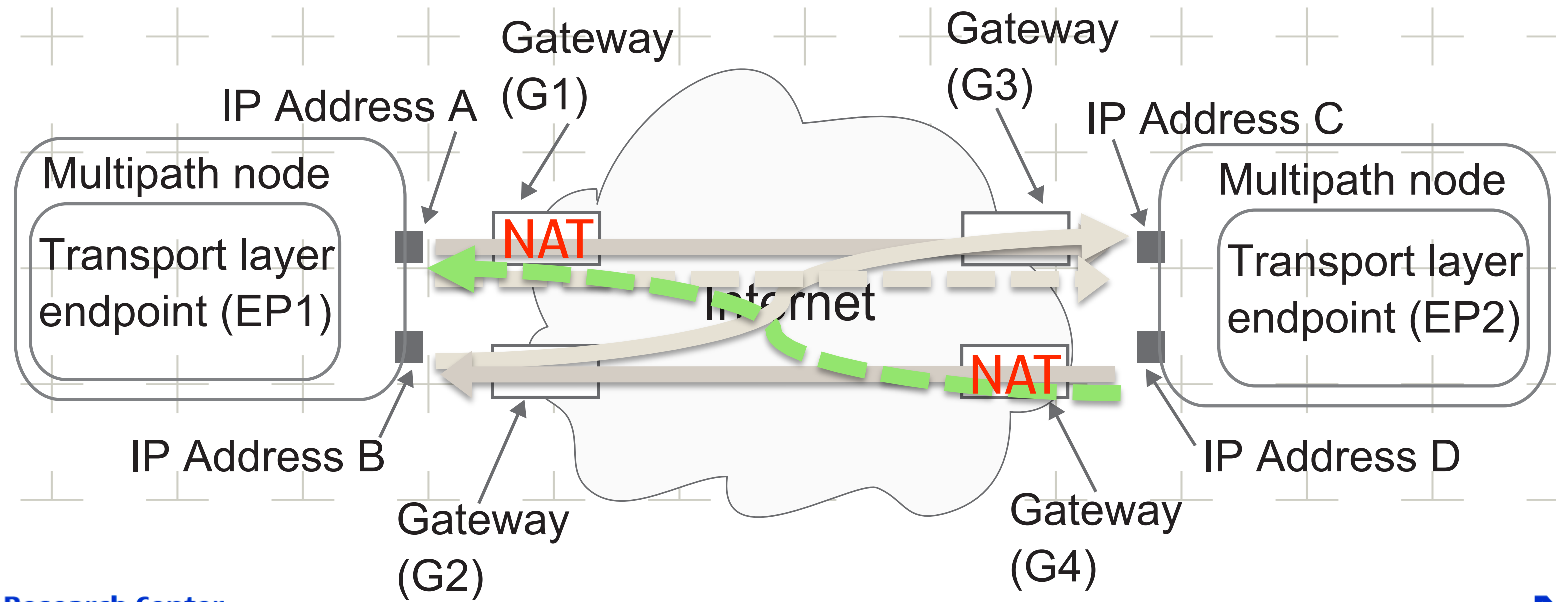
Nokia Research Center

NOKIA

Path discovery with subflow establishment

Subflows are also available between NAT boxes

- EP2 sends back a SYN/ACK from D (not from C)



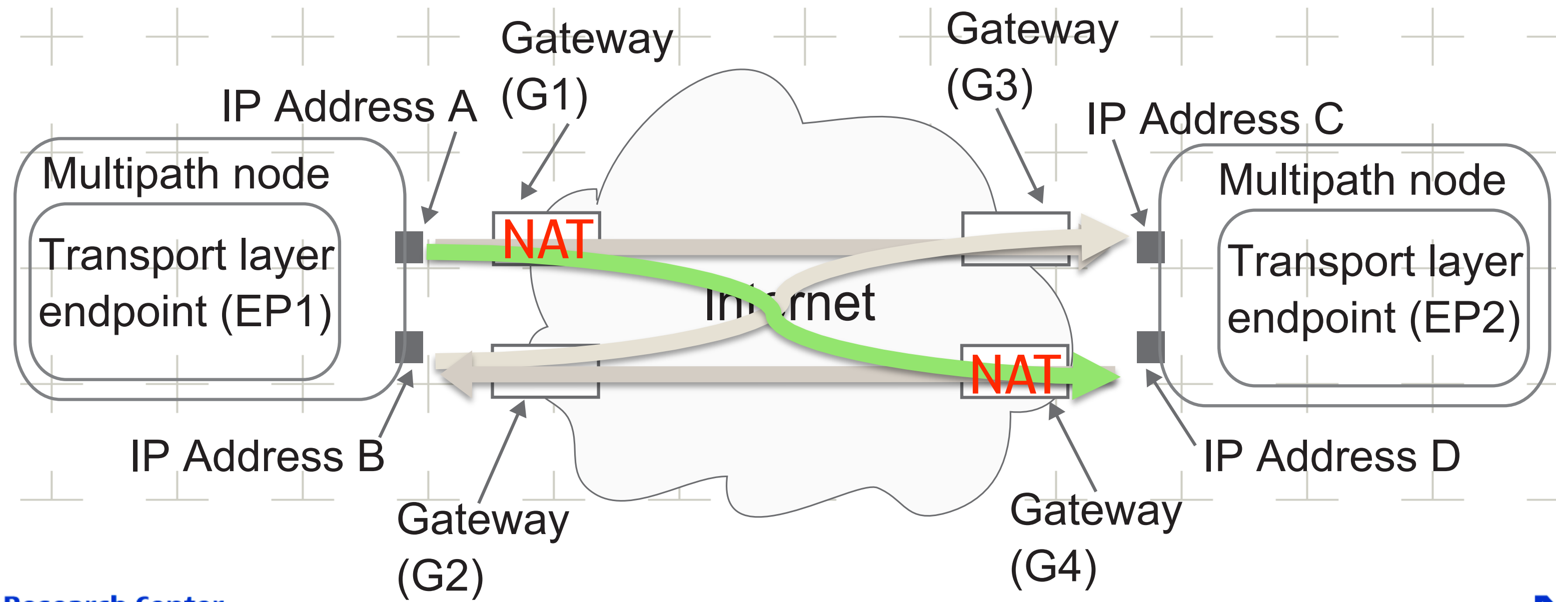
Nokia Research Center

NOKIA

Path discovery with subflow establishment

Subflows are also available between NAT boxes

- EP1 sends back an ACK from A to the SYN/ACK source



Nokia Research Center

NOKIA