

RIP (Routing Information Protocol): Spielanleitung

Ziel: Durch Zusammenarbeit mehrerer Gruppen füllen sich die Routingtabellen eines Routernetzes so, dass nach einigen Spieldurchläufen ein konvergiertes Netzwerk vorliegt.

Vorbereitung: Die Klasse wird in etwa gleichgroße Teams aufgeteilt. Jedes Team simuliert einen der Router in der Topologie. Dazu bekommt es einen Stapel mit vorab kopierten RIP-PDU-Zetteln (Vorlage in Datei „RIP-PDU.odt“) und eine Vorlage für die Routingtabelle (Datei „RIPRoutingProtokoll.ods“, Tabellenblatt „Routingtabelle“). An der Tafel wird in Form der Topologie für jeden Router eine Routingtabelle im A3-Format aufgehängt.

Ablauf: Den SuS wird der Aufbau einer Routingtabelle, der Algorithmus, nach dem RIP Routinginformationen verteilt und ausgewertet und der Aufbau einer RIP-PDU erklärt. (Struktogramm auf Seite 4 dieser Präsentation, die Originaldatei „RIPRoutingProtokoll.STR“ kann mit dem Tool „structuredit32“ bearbeitet werden)

Im ersten Durchlauf tragen alle Teams die direkt angeschlossenen Netzwerke in ihre RIP-PDU-Zettel (ein Zettel pro angeschlossenem Nachbarrouter), ihre eigene Routingtabelle und die Routingtabelle ihres Routers an der Tafel ein.

WICHTIG: die Metrik (Anzahl Hops bis zum Zielnetzwerk) wird auf den PDU-Zetteln um 1 gegenüber dem Eintrag in der Routingtabelle erhöht (aus Sicht des Nachbarn ist das publizierte Netz einen Hop weiter entfernt).

Wenn alle Teams die Eintragungen vorgenommen haben, wird zum ersten Routingupdate aufgerufen. Daraufhin verteilen sie die PDU-Zettel an ihre direkt angeschlossenen Nachbarn.

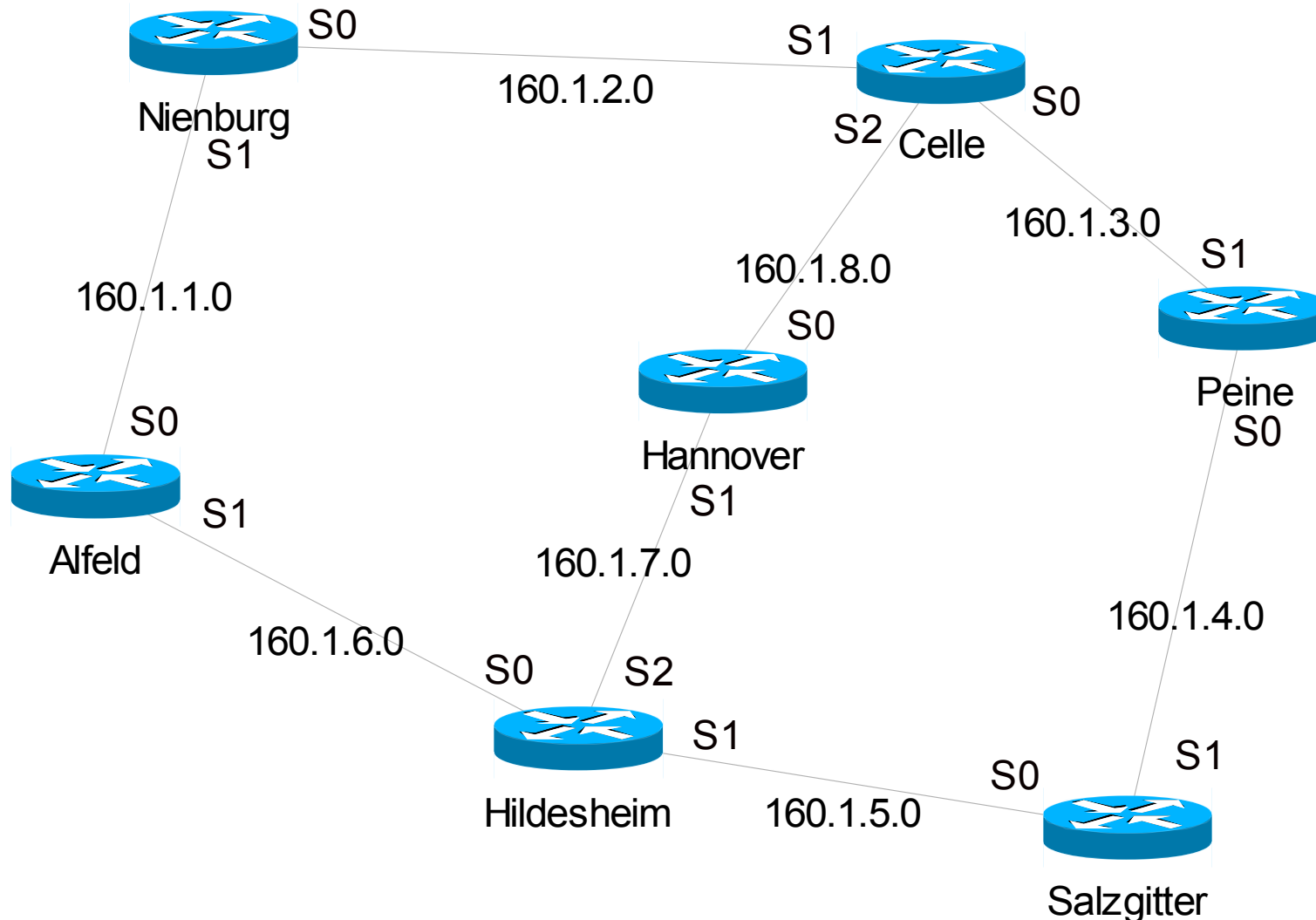
Die Informationen, die von den Nachbarn gesendet worden sind, werden nun gemäß dem RIP-Algorithmus in die eigene Routingtabelle eingetragen. (Neue Route besser --> alte Route streichen (wenn schon vorhanden) und neue eintragen; neue Route schlechter --> neue Route wird nicht eingetragen; neue Route gleich der alten: neue Route eintragen, alte behalten (load balancing)). WICHTIG: das outgoing interface ist immer jenes, von dem die Routinginformation erhalten wurde.

Nun wiederholen sich die Spieldurchläufe (Routingtabelle an der Tafel aktualisieren, RIP-PDU-Zettel ausfüllen, Routing Update) bis das Netz konvergiert ist.

Dauer: Ca. 90 min

Alternative: Ein Netzwerkpfad fällt aus, das Spiel wird wiederholt.

RIP (Routing Information Protocol): Topologie des Routernetzes



RIP: Aufbau einer Routing-Tabelle

Nienburg Routing Table						
Network		Metric	Outgoing Interface	Routing-Prot.	Update number	Comment
						Initial state: routing table is empty
Celle Routing Table						
Network		Metric	Outgoing Interface	Routing-Prot.	Update number	Comment
						Initial state: routing table is empty
Nienburg Routing Table						
Network		Metric	Outgoing Interface	Routing-Prot.	Update number	Comment
160.1.	2.0	0	S0	C	0	Fill in directly connected networks
160.1.	1.0	0	S1	C	0	
Celle Routing Table						
Network		Metric	Outgoing Interface	Routing-Prot.	Update number	Comment
160.1.	3.0	0	S0	C	0	Fill in directly connected networks
160.1.	2.0	0	S1	C	0	
160.1.	8.0	0	S2	C	0	

RIP: Algorithmus zum Aufbau einer Routing-Tabelle

RIP-Routing-Protokoll

Trage alle direkt angeschlossenen Netze in die Routing-Tabelle ein

Trage alle statischen Routen in die Routing-Tabelle ein

solange RIP als Routing-Protokoll aktiviert ist

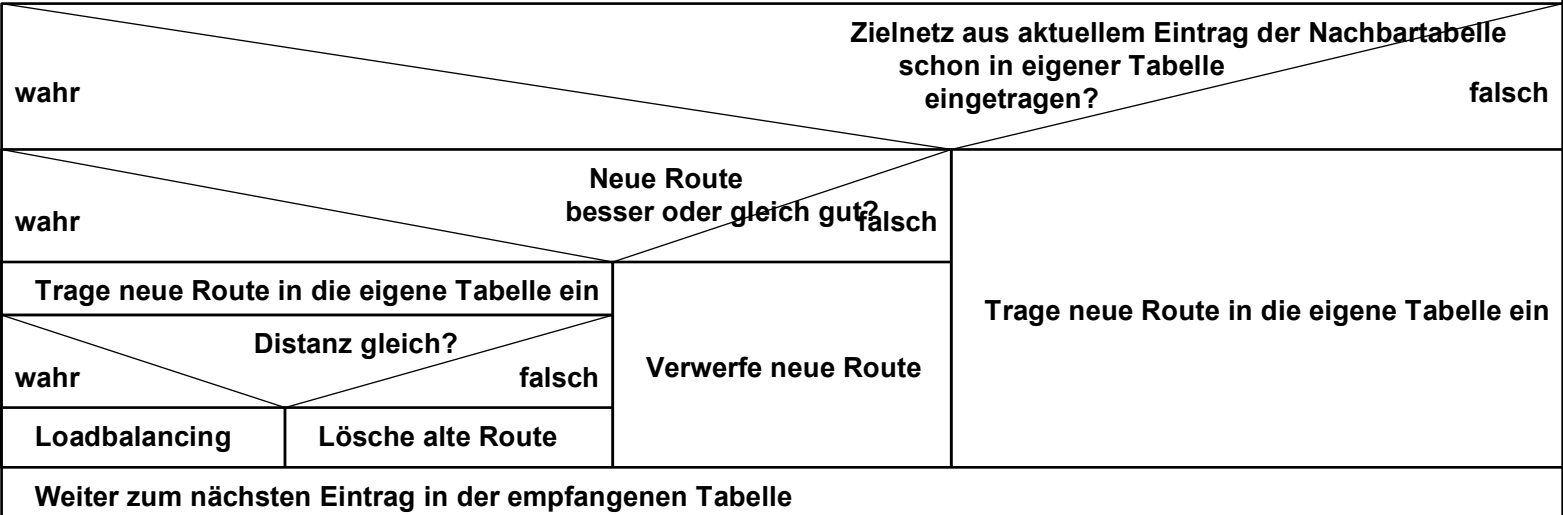
Erhöhe alle Hopanzahlen (Distanz) um 1

Sende diese Routing-Tabelle an alle direkten Nachbarn

Warte ...

(bis ein Routing-Update eines Nachbarrouters eintrifft) ODER
(bis die Update-Zeit abgelaufen ist)

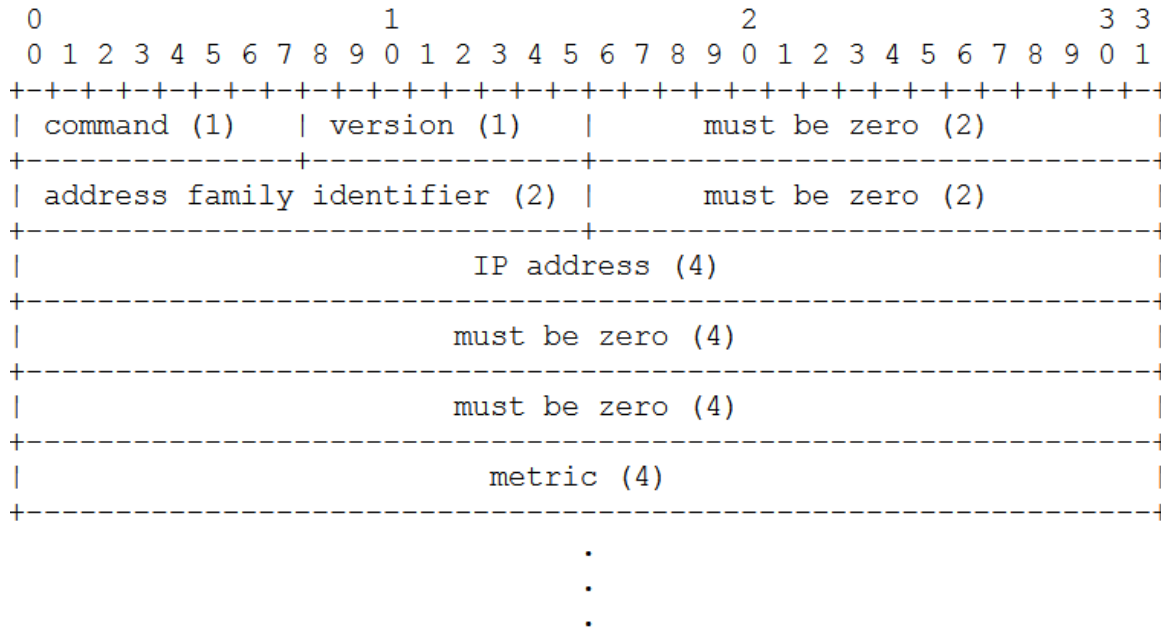
solange noch Routing-Einträge in der empfangenen Tabelle sind



RIP: Aufbau einer Routing-Tabelle, fortgeführt

Nienburg Routing Table						
Network		Metric	Outgoing Interface	Routing-Prot.	Update number	Comment
160.1.	2.0	0	S0	C	0	Received routing update from Celle
160.1.	1.0	0	S1	C	0	
160.1.	3.0	1	S0	R	1	Next Hop: Celle
160.1.	2.0	4	S0	R	4	Next Hop: Celle, Entry 1 is better --> delete this
160.1.	8.0	1	S0	R	1	Next Hop: Celle
Celle Routing Table						
Network		Metric	Outgoing Interface	Routing-Prot.	Update number	Comment
160.1.	3.0	0	S0	C	0	Received routing update from Nienburg
160.1.	2.0	0	S1	C	0	
160.1.	8.0	0	S2	C	0	
160.1.	2.0	4	S1	R	4	
160.1.	1.0	1	S1	R	1	
160.1.	3.0	2	S1	R	4	
160.1.	8.0	2	S1	R	4	

RIP: Aufbau der Protocol Data Unit und des Protocolstacks



The portion of the datagram from address family identifier through metric may appear up to 25 times. IP address is the usual 4-octet Internet address, in network order.

Quelle: RFC 1058

Every datagram contains a command, a version number, and possible arguments. This document describes version 1 of the protocol. Details of processing the version number are described in [section 3.4](#). The command field is used to specify the purpose of this datagram. Here is a summary of the commands implemented in version 1:

- 1 - request A request for the responding system to send all or part of its routing table.
- 2 - response A message containing all or part of the sender's routing table. This message may be sent in response to a request or poll, or it may be an update message generated by the sender.

Quelle: Wikipedia

RIP im TCP/IP-Protokollstapel:

Anwendung	RIP				
Transport	UDP				
Internet	IP (IPv4, IPv6)				
Netzzugang	Ethernet	Token Bus	Token Ring	FDDI	...

RIP: Protocol Data Unit für das Rollenspiel

<u>RIP Routing update</u>		
received from: _____		
received at: _____:_____h		
command 2	version 1	must be zero
address family identifier 2		must be zero
IP-address		
must be zero		
must be zero		
metric		
IP-address		
must be zero		
must be zero		
metric		
IP-address		
must be zero		
must be zero		
metric		
IP-address		