

openHPI-Kurs: Wie funktioniert das Internet?
Wichtige Routingverfahren

Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut
Universität Potsdam

Isoliertes Routing (1/3)

Beim **isolierten Routing** werden nur die lokal verfügbaren Informationen, z.B. Länge der Warteschlange vor einem Ausgang, oder Zustand eines Nachbarn ausgewertet

Wichtige Beispiele:

- Fluten, selektives Fluten oder Hot-Potato

Fluten oder Flooding:

- **Idee:** Jedes eintreffende Paket wird über alle anderen Anschlüsse des Routers weitergeleitet
 - Vervielfachung der Pakete erzeugt sehr hohe Netzlast
 - Zur Eindämmung werden den Paketen Hop-Counter oder Zeitstempel mitgegeben und es wird sichergestellt, dass ein Router ein Paket nie zweimal ausliefert

Isoliertes Routing (2/3)

Selektives Fluten oder **Selective Flooding**:

- **Idee**: Jedes eintreffende Paket wird über alle Anschlüsse des Routers weitergeleitet, die in etwa in die richtige Richtung weisen
 - Router brauchen dazu Informationen über den Aufbau des Netzwerks

Fluten und selektives Fluten sind wegen Erzeugung hoher Netzlast meist nicht praktikabel

Anwendungen sind jedoch sinnvoll bei

- Verteilung von Updates für verteilte Datenbanken (Synchronizität)
- militärischen Anwendungen mit hoher Störanfälligkeit

Aber **Evaluation**: Fluten findet stets die kürzeste Route

Isoliertes Routing (3/3)

Hot-Potato-Routing:

- **Idee:** Router versucht, eintreffendes Paket auf dem schnellsten Weg loszuwerden, also über den Ausgang mit der aktuell kürzesten Warteschlange
- **Probleme:**
 - Pakete finden selten kürzesten Weg, sondern machen Umwege
 - Verstärkung von Netzwerküberlastung – Pakete werden abgesendet, auch wenn sich Pakete auf dem Weg zum Empfänger stauen

Distanzvektor-Routing (1/2)

Distanzvektor-Routing (oder Bellman-Ford- oder Ford-Fulkerson Algorithmus) ist dezentrales dynamisches Routing-Verfahren

- Jeder Router berechnet seine Routing-Tabelle lokal und teilt Nachbar-Router ein in seinen Kenntnisstand von der Netzwerktopologie
- Routing-Informationen werden periodisch ausgetauscht, sodass nach einer gewissen Zeit jeder Router die kürzesten Wege zu allen Zielen im Netzwerk kennt
- Durch den periodischen Informationsaustausch können sich Router an Veränderungen im Netzwerk anpassen, z.B. bei Ausfall einer Verbindung
- Jeder Router speichert (in einem Vektor) neben Next-Hop-Eintrag auch die Distanz zum Ziel → **Distanzvektor**

Distanzvektor-Routing (2/2)

- Distanzberechnung erfolgt auf Basis einer vorgegebenen Routing-Metrik, z.B. Anzahl der Hops, Warteschlangenlänge, Übertragungszeit, ...
- Distanzvektor (DV) wird durch Austausch von Routing-Informationen mit den Nachbar-Routern ständig aktualisiert
 - Router prüfen Dreiecksungleichung:
Distanz zum Ziel Z in DV von Nachbar A + ...
... + Distanz zu A $\leq$ Distanz zu Z in eigenem DV?
→ eigenen DV updaten
- Distanzvektor-Routing war das erste Routing-Verfahren im ARPANET, es wurde in RFC 1058 als **Routing Information Protocol – RIP** standardisiert

OSPF – Link-State-Routing

Distanzvektor-Routing-Verfahren wurde im Internet wegen seiner langsamen Konvergenz durch **OSPF – Link-State-Routing** – abgelöst

Link-State-Routing ist dezentrales dynamisches Routing-Verfahren

- Router tauschen Informationen über Status der Verbindungen zu Nachbarn aus und keine vollständige Tabellen
- Routing-Informationen werden per Broadcast gleichzeitig an alle Router im Netzwerk gesendet
- Verfahren erlaubt sehr schnelle Anpassung an sich verändernde Netzwerksituationen
- Standardisiert in RFC 2328
- **OSPF** („Open Shortest Paths First“) basiert auf „Shortest Paths First“-Algorithmus aka. **Dijkstra-Algorithmus**