

Vorlesung

# Advanced Networking Technologies

Dr. Michael Roßberg

# Inhaltsverzeichnis

<b>1. Routers and Switches</b>	<b>3</b>
1.1. Why we Need Faster Routers . . . . .	3
1.2. Why Making Fast Routers is Difficult . . . . .	3
1.3. First Generation Routers . . . . .	3
1.4. Second Generation Routers . . . . .	4
1.5. Third Generation Routers . . . . .	4
1.6. Fourth Generation Routers . . . . .	4
1.7. Generic Router Architecture . . . . .	4
1.8. Buffer Placement . . . . .	5
1.8.1. Output queueing . . . . .	5
1.8.2. Input queueing . . . . .	5
1.8.3. Virtual Output Queueing . . . . .	5
<b>A. Buzzword Of The Day</b>	<b>6</b>
A.1. Cut-Through-Switching . . . . .	6

# 1. Routers and Switches

Router sind zuständig dafür, Pakete anhand der Informationen im IP-Header (i. d. R.) weiterzuleiten. Das passiert anhand der Forwarding-Tabelle (berechnet aus der Routing-Tabelle), die eindeutig einen Next Hop für eine Zieladresse festlegt. Router werden üblicherweise in *Points of Presence* (POPs) (z. B. DE-CIX) zusammengeschlossen. Das Internet ist also kein gut vermaschtes Netz, sondern die Vermaschung konzentriert sich eher auf wenige POPs.

## 1.1. Why we Need Faster Routers

Router werden i. d. R. für große Bandbreiten ausgelegt, da diese sonst leicht zum Bottleneck werden können. Schnelle Router sind wichtig, um Kapazitäten, Kosten, Größe und Stromverbrauch am POP zu senken. Entscheidend sind die Port-Kosten. Je weniger Ports benutzt werden sollen, umso schneller muss der Router werden. In POPs mit großen Routern können Pakete innerhalb des POPs mit deutlich weniger Interconnections weitergeleitet werden, während bei kleineren Routern deutlich mehr Verbindungen nötig sind (um alles miteinander zu verbinden).

Zur Steigerung von Übertragungsgeschwindigkeiten ist es bspw. auch möglich, die Wandlung der Daten in elektrische Signale an Routern für einzelne Farben in einem WDM auszulassen und stattdessen über ein Prisma die Farbe direkt an die passende Zielfaser weiterzuleiten. Dies spart teure Umwandlungen und erhöht den Durchsatz.

## 1.2. Why Making Fast Routers is Difficult

Moore's Law ist für CPUs nicht mehr gültig und wurde für Speicher nie erreicht. Weder Kapazität, Bandbreite noch Zugriffszeiten bei Speicher folgen Moore's Law, sondern steigen deutlich langsamer.

## 1.3. First Generation Routers

Zu BNC-Zeiten gab es für jeden Anschluss ein Line Interface mit BNC-Anschlüssen, die an einer gemeinsamen Backplane angeschlossen waren. Durch die Bus-Architektur muss jedes Pakete zweimal über den Datenbus gesendet werden.

## 1.4. Second Generation Routers

Zusätzlich werden auf den Line Cards Forwarding Caches eingebaut, die ausgehende Interfaces zwischenspeichern, wodurch die meisten Pakete nur einmal über den Datenbus geschickt werden müssen. Durch das Caching können jedoch Reordering-Probleme auftreten, wenn ein zweites Paket dank Cache schon verschickt werden kann, während das erste Paket noch im Paketpuffer ist.

## 1.5. Third Generation Routers

In der Backplane wird eine Switchingmatrix gepflegt, um Pakete hin- und herzusenden. In den Line Cards wird jeweils eine Forwarding-Tabelle von der CPU gepflegt, sodass keine Zugriffe mehr auf die Backplane nötig sind, um das Zielinterface zu bestimmen.

## 1.6. Fourth Generation Routers

Router sind teilweise als Multi-Chassis-Systeme mit optischen Links zwischen den Chassis aufgebaut. Damit hat man im Prinzip schon ein Netzwerk innerhalb des Routers.

## 1.7. Generic Router Architecture

Bei jedem eingehenden Paket wird anhand der IP-Adresse der Zielport aus der Forwarding-Table ausgelesen. Header (TTLs) werden aktualisiert und über ein Switching Fabric an den Output Buffer des ausgehenden Interfaces geschickt. Anschließend wird das Paket am ausgehenden Link verschickt.

Für diese Vorgänge ist generell sehr wenig Zeit möglich; bspw. sind für 40 Gbps-Switching 8 ns Zeit für IP-Adress-Lookup verfügbar. Solche Lookups können jedoch nicht mit einfachen Hash-Tabellen gelöst werden, da diese Worst Case mehr Speicher nehmen als verfügbar ist. Durch Bäume lassen sich zwar die hierarchischen Strukturen, die für Lookups in CIDR nötig sind, speichern, jedoch sind bei Baumsuchen Speicherzugriffe nicht sehr cacheeffizient, was Lookups durch Cache Misses sehr teuer machen kann.

Für die Speicherung im Router werden statt normalem RAM TCAMs benutzt. Dabei werden die Adressen für die Routen hinterlegt. Alles hinter der Subnetzmaske wird dabei auf „don't care“ gesetzt. Für eingehende Pakete werden dann einfach alle Lines im TCAM parallel durchsucht und der Eintrag mit der höchsten Priorität benutzt. So kann deterministisch und schnell eine Pattern-Suche gemacht werden. Nachteil dieser Technik sind hoher Energieverbrauch (es wird immer der ganze Speicher angesprochen) und hohe Kosten.

Für die Logik werden gerne FPGAs/ASICs eingesetzt.

## 1.8. Buffer Placement

Pakete müssen gelegentlich auch mal gespeichert werden. Dies kann entweder am Input Port oder am Output Port gemacht werden. Je nachdem, wo man dies tut, hat es verschiedene signifikant Eigenschaften.

### 1.8.1. Output queueing

Output queueing ist hinsichtlich der Latenz optimal. Worst Case kommt an jedem Input Port etwas für denselben Output Port an, die jedoch dort einfach gespeichert werden können, ohne andere Prozesse zu stören. Allerdings muss das Switching Fabric das  $n$ -fache ( $n$  ist die Anzahl Ports) der Line Rate schaffen.

### 1.8.2. Input queueing

Queues werden kurz vor dem Switching Fabric eingesetzt. Wenn jetzt also mehrere Ports einen Output Port ansprechen wollen, müssen die meisten warten. Für die Entscheidung, welches Paket wann geschickt wird, ist jetzt ein Scheduler nötig. Dafür muss die Switching Fabric aber nur so viel Durchsatz wie Line Rate haben, da an jeden Output Port höchstens mit Line Rate gesendet wird.

Input queueing ist anfällig für Head of Line Blocking. Wenn ein Paket in der Queue nicht verschickt werden kann, können auch nachfolgende Pakete dieses Input Ports (also auch solche, die an andere, freie Ports gerichtet sind), nicht verschickt werden. Also erhöht sich die Latenz und der maximale Durchsatz verringert sich.

### 1.8.3. Virtual Output Queueing

An die Input Queues werden jetzt mehrere Queues gesetzt (eine für jeden Output Port). Jetzt kann ein Scheduler entscheiden, welcher Port jetzt für welchen Output Port dran kommt. Damit wird es wieder Latenzoptimal.

# A. Buzzword Of The Day

In den Vorlesungen wird ein „Buzzword“ bzw. Begriff aus der Welt der Netzwerktechnik genannt. Dieser Begriff ist von den Hörenden zu recherchieren und wird in der anschließenden Vorlesung diskutiert. Hier sind einige Begriffe samt Diskussionen und zusammengetragen.

## A.1. Cut-Through Switching [1, 3]

- Was bedeutet das?
- Ist das eine sinnvolle Technologie?

Cut-Through Switching vermeidet Latenz, indem ein Paket bereits an den ausgenden Port verschickt wird, sobald die Zieladresse und der ausgehende Port bestimmt werden konnten. Das Paket kann also weitergeleitet werden, noch bevor es vollständig empfangen wurde.

Cut-Through Switching erfordert, dass die Linkgeschwindigkeit des ausgehenden Ports mindestens so groß ist wie die des eingehenden Ports, da sonst das Paket wieder gepuffert werden muss. Ein weiteres Problem ergibt sich dadurch, dass die Checksumme des Ethernet Frames nicht geprüft werden kann, bevor das Paket verschickt wird (einerseits, da die Payload zu diesem Zeitpunkt noch nicht ganz bekannt ist, andererseits, da die Checksumme erst am Ende des Ethernet Frames steht). Daher leitet ein Cut-Through Switch auch beschädigte Frames weiter, die ein Store-and-Forward Switch verwerfen würde.

Cut-Through Switching kann auch in einer adaptiven Variante betrieben werden. In diesem Fall wird es nur benutzt, solange Fehler am Link nur selten passieren. Steigt hingegen die Fehlerrate am Port, wird auf Store and Forward umgeschaltet, um unnötiges Weiterleiten von Paketen zu vermeiden.

Eine andere Erweiterung ist Fragment-Free-Cut-Through, bei der nur die ersten 64 Bytes eines Ethernet Frames geprüft werden. Dies beruht auf der Erfahrung, dass Framefehler meistens nur innerhalb der ersten 64 Byte auftreten.

# Literatur

- [1] *Cut-through switching*. In: *Wikipedia*. Page Version ID: 1205281012. 9. Feb. 2024.  
URL: [https://en.wikipedia.org/w/index.php?title=Cut-through\\_switching&oldid=1205281012](https://en.wikipedia.org/w/index.php?title=Cut-through_switching&oldid=1205281012) (besucht am 08.04.2024).
- [2] Michael Roßberg. “Advanced Networking Technologies”. Vorlesung. Vorlesung. 2024.
- [3] *Switching*. *Elektronik-Kompendium*. URL: <https://www.elektronik-kompendium.de/sites/net/0907141.htm> (besucht am 08.04.2024).