Vorlesung Kommunikationssysteme
Wintersemester 2026/27

LAN Komponenten
und WAN Technologien

Christoph Lindemann

Zeitplan

Nr.	Datum	Thema
01	16.10.26	Organisation, Internet Trends und Grundlagen von Android
02	23.10.26	Protokolldesign und das Internet
03	30.10.26	Anwendungen und Netzwerkprogrammierung
04	06.11.26	LAN und Medienzugriff
05	13.11.26	Ethernet und drahtlose Netze
06	20.11.26	LAN Komponenten und WAN Technologien
07	27.11.26	Routingverfahren und Protokolle
08	04.12.26	Internetworking und Adressierung mit IP
09	11.12.26	IP Datagramme
10	18.12.26	Zusätzliche Protokolle und Technologien
11	08.01.27	User Datagram Protocol und Transmission Control Protocol
12	15.01.27	TCP Überlastkontrolle / Internet Routing und Routingprotokolle
13	22.01.27	TCP für Hochgeschwindigkeitsnetze
14	29.01.27	Review der Vorlesung

Überblick

Ziele:

- Überblick über die Hardwarebausteine eines LANs
- Vertiefung des Verständnisses von WAN Technologien

Themen:

- LAN-Komponenten
  - Repeater
  - Bridge
  - Switch
- WAN-Technologien
  - Architektur
  - Routing
- Sonstige Technologien

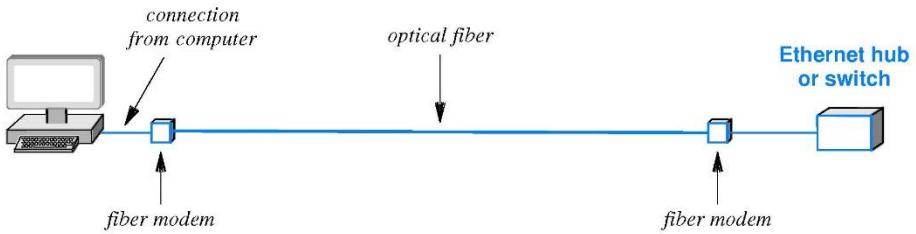
LAN-Komponenten

Distanzen in LANs

- Hardware sendet feste Menge an Energie
- Überschreitet Länge der Verkabelung bestimmte Grenzen, empfängt Station kein genügend starkes Signal → Fehler
- Deswegen:
  - LAN Technologie enthält Beschränkung der Länge
  - Keine korrekte Funktion der Technologie bei deren Überschreitung

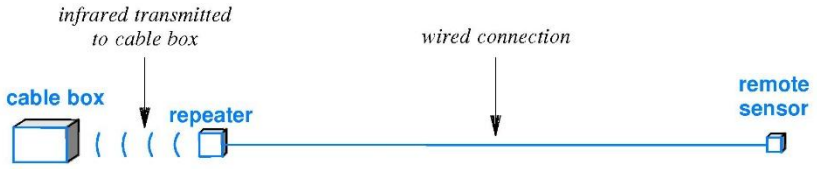
Fiber Modems

- Anbindung eines Hosts an entferntes Ethernet
- Glasfaserleiter zwischen zwei Fiber Modems
- Transparent für Computer und Ethernet Switch falls Modem Standard Interface verwendet
- In Praxis zwei Glasfaserleiter für simultane Übertragung in beide Richtungen



Repeater

- Analoges Gerät um LAN Signale über weite Entfernungen zu übertragen
- Empfanges Signal wird verstärkt und weiter gesendet
- Kein Verständnis von Paketen oder Bits
- Früher häufig im Ethernet genutzt
- Auch: Weiterleitung von Infrarotsignalen



Erweiterung von Infrarotsensor mit Repeater

Bridges

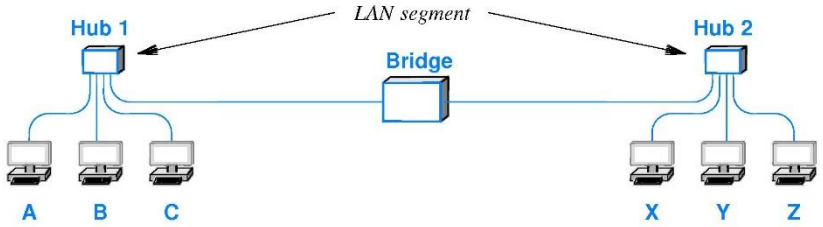
- Verbindung und Paketübertragung zwischen zwei LANs
- Bridge lauscht auf beiden Segmenten in Promiscuous Mode (Empfängt alle Pakete)
- Bridge empfängt valides Paket auf einem Segment → Weiterleitung an anderes Segment
  - Broadcast Frame an alle Computer beider Segmente
- Für Computer nicht erkennbar, ob einziges LAN oder LANs mit Bridge
- Früher Stand-Alone Gerät, heute in Kabel- oder DSL-Modem sowie Wireless Router integriert

Lernende Bridges und Filterung (1)

- Bridge analysiert Ziel MAC Adresse des Frames
- Unnötige Weiterleitungen werden verhindert → **Filterung**
- Broadcast und Multicast werden immer weitergeleitet
- **Adaptive / Lernende** Bridges lernen Orte der Computer automatisch
- Source MAC Adresse wird extrahiert und zu Liste für Segment hinzugefügt

Lernende Bridges und Filterung (2)

- Beispiel von zwei LAN Segmenten mit einer Bridge



LAN mit Bridge und sechs Computern

- Veränderung der Segmentlisten nach bestimmten Ereignissen

Event	Segment 1	Segment 2	Frame Travels
Bridge boots	–	–	–
A sends to B	A	–	Both Segments
B sends to A	A, B	–	Segment 1 only
X broadcasts	A, B	X	Both Segments
Y sends to A	A, B	X, Y	Both Segments
Y sends to X	A, B	X, Y	Segment 2 only
C sends to Z	A, B, C	X, Y	Both Segments
Z sends to X	A, B, C	X, Y, Z	Segment 2 only

Segmentlisten und Frameübertragungen

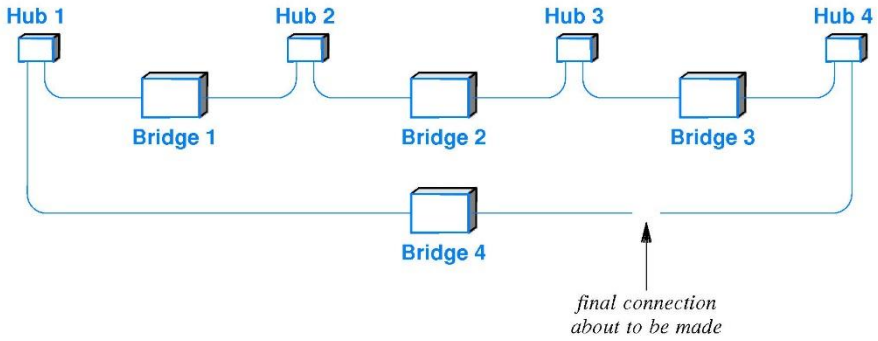
- Entscheidung, ob Frame in anderes Segment übertragen wird

Performance einer Bridge

- Nach Lernphase bessere Performance als einzelnes LAN erreichbar
- Simultane Übertragungen in beiden Segmenten möglich
- Bridge zwischen Gebäuden
  - Computer kommuniziert eher mit nahem Drucker statt entferntem Drucker
- Bridge zwischen ISP und Kunde in DSL- oder Kabelmodem
  - Isoliert Netzwerk des Kunden von Netzwerk des ISP
  - Lokale Kommunikation wird nicht an ISP geleitet

Distributed Spanning Tree (1)

- Bridge 4 soll eingefügt werden
- Netzwerk funktioniert vorher wie erwartet
  - Broadcast/Multicast wird von Bridge an anderes Segment weitergeleitet
- Mit Bridge 4 entsteht eine Schleife → Broadcast wird endlos weitergeschickt



Distributed Spanning Tree (2)

- Bridges berechnet **Distributed Spanning Tree (DST)**
- Bildet einen Graphen: Bridge als Knoten, verbunden als Baum
- **Spanning Tree Protocol (STP):** Ursprünglicher Ansatz von DEC 1985
- Drei Schritte:
  - Wahl der Wurzel
  - Berechnung des kürzesten Pfad
  - Weiterleitung

- Bridges kommunizieren über reservierte Multicast Adresse 01:80:C2:00:00:00

Distributed Spanning Tree (3)

- Wahl der Wurzel
  - Bridge sendet ID als Multicast (16 Bit Priorität + 48 Bit MAC Adresse)
  - Bridge mit kleinster ID wird genommen
- Berechnung des kürzesten Pfad
  - Jede Bridge berechnet kürzesten Pfad zu Wurzel
  - Verbindungen in kürzestem Pfad aller Bridges bilden Spanning Tree
- Weiterleitung
  - Nur Interface mit Verbindung zu kürzestem Pfad darf weiterleiten

Weitere DST Algorithmen

- **802.1d:** 1990 von IEEE entwickelt, 1998 aktualisiert
- **802.1q:** Spanning Tree auf logisch unabhängigen Netzwerken mit gemeinsamem, physischem Medium
- **802.1w:** 1998 von IEEE, Einführung von Rapid Spanning Tree Protocol mit verbesserter Geschwindigkeit nach Topologieänderung
- **801.1d-2004:** Mit Rapid Spanning Tree, ersetzt STP
- Zusätzlich: Per-VLAN Spanning Tree (PVST), PVST+, Multiple Instance Spanning Tree Protocol (MISTP), Multiple Spanning Tree Protocol (MSTP)

Switching und Layer 2 Switch (1)

- **Ethernet Switch (Level 2 Switch)**

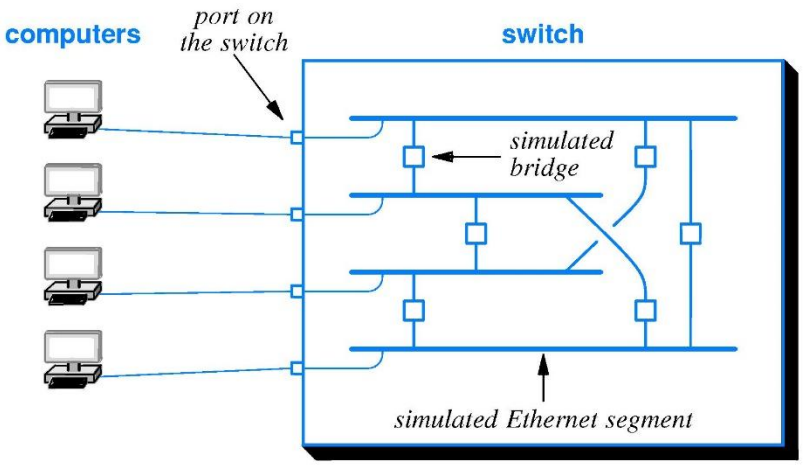
- Ähnlich zu einem Hub
  - Mehrere Ports für einzelne Computer
  - Computer kann Frame an anderen Computer am Switch senden

- Hub ist analoges Gerät und leitet Signale weiter
  - Simuliert geteiltes Übertragungsmedium

- Switch ist digitales Gerät und leitet Pakete weiter
  - Simuliert Netzwerk mit Bridges und einem Computer pro Segment

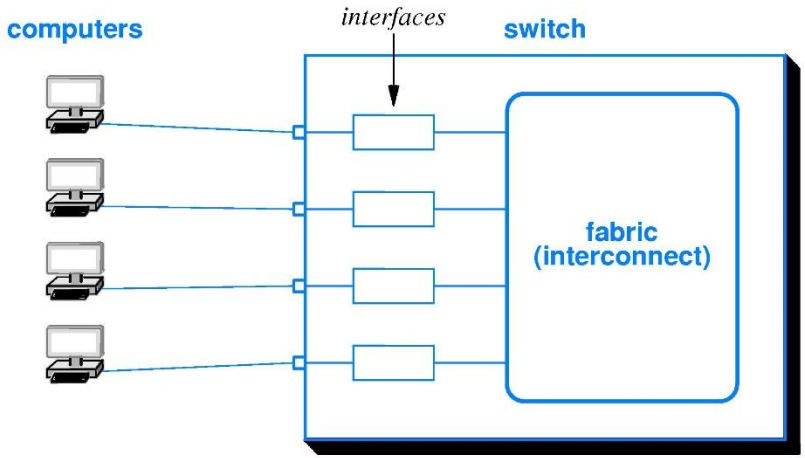
Switching und Layer 2 Switch (2)

- Konzept eines Switches
- Entspricht nicht tatsächlichem Aufbau des Switch



Switching und Layer 2 Switch (3)

- Jeder Port mit Intelligent Interface (enthält Prozessor, Speicher, andere Hardware)
  - Kann Pakete in Speicher puffern, falls Output Port beschäftigt
- Zentrale Fabric: Erlaubt simultane Übertragungen zwischen jeweils zwei Interfaces



Switching und Layer 2 Switch (4)

- Wichtigster Vorteil gegenüber HUB ist parallele Übertragung
- Übertragungen müssen unabhängig sein (nur ein Paket an einen Port zu Zeitpunkt)
- Switch mit N Ports und N Computern schafft N/2 Übertragungen gleichzeitig
- Switches können zwischen wenigen Ports (z.B. 4 für Heimnetzwerk) und über 1000 Ports (Firmen, ISPs) variieren

Zusammenfassung

- Mehrere Mechanismen um LANs über größere Entfernungen aufzubauen
- Fiber Modem um entfernten Computer mit LAN zu verbinden
- Repeater verstärkt elektrische Signale
- Bridge verbindet zwei LAN Segmente und überträgt Pakete
  - Lernt Zugehörigkeit zu Segmenten und filtert Pakete
- Switch verbindet mehrere Computer untereinander
  - Erlaubt im Gegensatz zu Hub parallele Übertragungen
  - VLAN Switch simuliert mehrere Switches

WAN-Technologien und Dynamisches Routing

WAN (1)

- Netzwerktechnologien können nach der überbrückten Distanz klassifiziert werden
  - PAN: Region um eine Person
  - LAN: Gebäude oder Campus
  - MAN: Gebiet einer großen Stadt
  - WAN: Mehrere Städte oder Länder

WAN (2)

- WAN oder LAN?
  - Firma mit Satellit-Bridge zwischen zwei LANs nur ein erweitertes LAN
- WAN und LAN unterscheiden sich in Skalierbarkeit
- WAN muss bei Bedarf wachsen können um viele geographische Standorte zu verbinden
- WAN-Technologie muss angemessene Performance für großes Netzwerk liefern

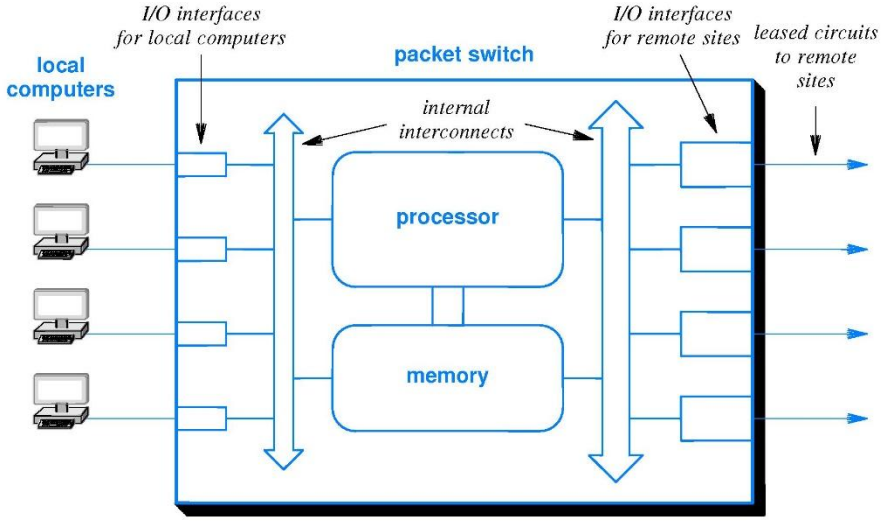
Ursprüngliche WAN Architektur (1)

- Moderne Kommunikationssysteme überbrücken große Distanzen mit Internettechnologie
  - Verbindung über Router an jedem Standort

- Hier: Ein einzelnes Netzwerk über große Distanzen
- Entwicklung begann vor Internet und LANs als Long-Haul-Networks
- Zu Beginn Verbindungen zwischen wenigen Computer an vielen Standorten
- **Packet Switch**: Lokale Verbindungen für Computer am Standort sowie Verbindungen zu Datenleitungen zu anderen Standorten

Ursprüngliche WAN Architektur (2)

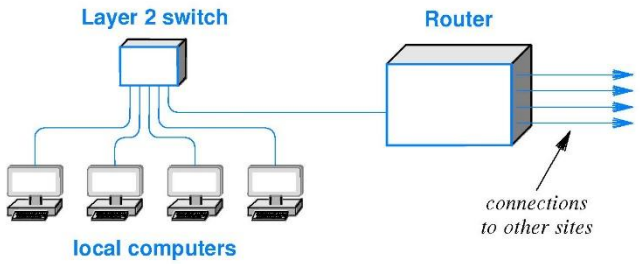
- Ursprünglich konventionelle Computer, später Spezialhardware
- Hochgeschwindigkeits I/O zu anderem Switch über gemietete Leitung
- Langsamere I/O zu lokalen Computern



Ursprüngliche Packet Switch Architektur

Ursprüngliche WAN Architektur (3)

- Seit LANs wird Packet Switch in WAN in zwei Teile unterteilt
  - Layer 2 Switch um lokale Computer zu verbinden
  - Router um zu anderen Standorten zu verbinden



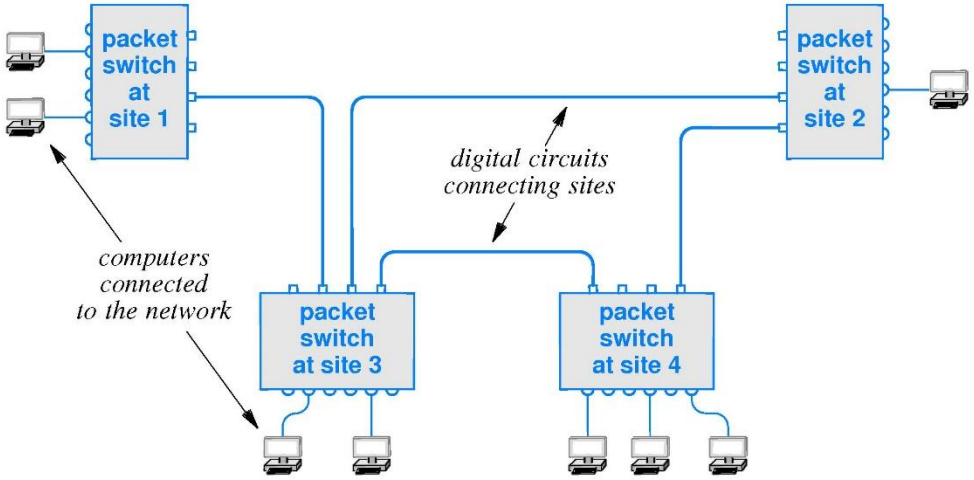
Modernes WAN mit lokaler Kommunikation in separatem LAN

Aufbau eines WAN (1)

- WAN gebildet durch Verbindungen von Packet Switches an mehreren Standorten
- Details hängen von benötigter Datenrate, Distanz und tolerierter Verzögerung ab
- Nutzen oft gemietete Standleitungen (T3, OC-12, …)
- Auch möglich: Mikrowellen, Satellitenkanal

Aufbau eines WAN (2)

- Topologie muss gewählt werden
- WAN muss nicht symmetrisch aufgebaut sein
- Verbindungen und deren Kapazität nach Bedarf gewählt (erwarteter Verkehr, Redundanz)



Beispiel eines WAN mit verbundenen Packet Switches

Store and Forward (1)

- **Store and Forward**: Fundamentales Paradigma um simultane Übertragungen zu erlauben
- Packet Switch puffert Pakete im Speicher
- Store: I/O Hardware im Packet Switch speichert Kopie des Paket im Speicher
- Forward: Prozessor analysiert Paket, bestimmt Ziel, sendet Paket über I/O Interface
- Pakete zu selben Ausgabegerät werden gespeichert bis Gerät bereit

Store and Forward (2)

- Vorheriges Beispiel: Zwei Computer an Standort 1 senden gleichzeitig Paket für Computer an Standort 3
- I/O Hardware legt Paket in Arbeitsspeicher und informiert Packet Switch Prozessor
- Prozessor bestimmt Standort 3 als Ziel
- Ist Output Interface nicht beschäftigt startet Übertragung direkt
- Sonst Speicherung des Paket in Warteschlage des Output Interface bis es frei ist

Adressierung im WAN

- WAN Technologie definiert Frame Format, welches Computer zum Senden / Empfangen nutzt

- Jeder Computer des WAN bekommt Adresse, bei Versand muss Zieladresse angegeben werden

- WANs nutzen Hierarchische Adressierung
  - Konzept: (Standort, Computer des Standort)
  - Packet Switch an Standort hat eindeutige ID, bestimmt ersten Teil der Adresse
  - Adresse ist tatsächlich ein Binärwert



Next-Hop Forwarding (1)

- Packet Switch muss ausgehenden Pfad für Paket bestimmen
- Software in Packet Switch analysiert Zieladresse
- Selbe Switch ID: Paket für lokalen Computer wird direkt zu Ziel gesendet
- Andere Switch ID: Weiterleitung über eine Verbindung zu anderem Switch
- Benötigt keine Information wie jeder einzelne Computer erreicht wird
- Muss nicht komplette Route durch Netzwerk berechnen

Next-Hop Forwarding (2)

- **Gegeben**: Angekommenes Paket an Switch Q
- **Durchführung**: Next-Hop Forwarding
- **Verfahren**:

  Extrahieren der Zieladresse aus Paket

  Teilen der Adresse in Packet Switch ID P und Computer ID C

  if (P == Q) /* lokales Ziel */

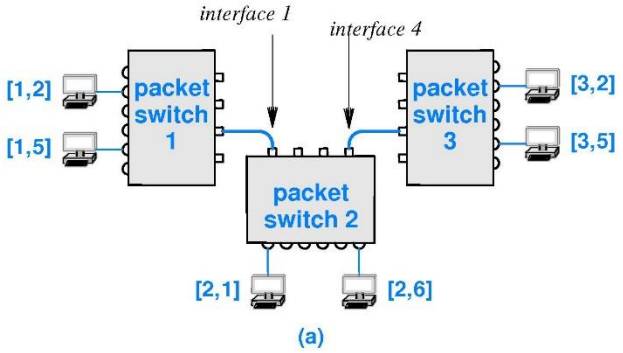
  Weiterleitung an lokalen Computer C

  else

  Wähle Verbindung zu anderem Packet Switch und leite darüber weiter

Next-Hop Forwarding (3)

- Switch nutzt ersten Teil der Adresse um nächsten Hop zu Ziel zu berechnen
- Verwendet Weiterleitungstabelle mit allen möglichen Packet Switches und nächsten Hop für jeden
- Schnelle Berechnung: Ein Eintrag pro Switch statt Computer, Organisation der Tabelle als Array statt komplette Suche



to reach	send to
switch 1	interface 1
switch 2	local delivery
switch 3	interface 4

(b)

(a) Netzwerk mit drei Packet Switches
(b) Next-Hop Weiterleitungstabelle

Quellenunabhängigkeit

- Next-Hop Forwarding unabhängig von Quelle oder genommenen Pfad des Paket
- Next-Hop hängt nur von dem Ziel ab
- Dadurch ist Weiterleitung kompakt und effizient
  - Nur Zieladresse muss extrahiert werden
  - Eine Tabelle reicht, da alle Pakete selben Pfad folgen
  - Pakete von direkt verbundenen Computern sowie anderen Switches nutzen selben Mechanismus

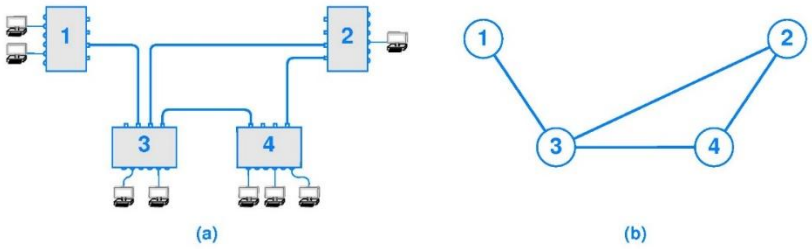
Dynamisches Routing im WAN (1)

- Werte in Weiterleitungstabelle garantieren:
  - Universelle Kommunikation: Gültige Next-Hop Route zu jeder möglichen Zieladresse
  - Optimale Route: Next-Hop für Ziel muss zu kürzestem Pfad zu Ziel führen
- Änderung des Pfades falls Netzwerkfehler vorhanden ist
- Routing Software auf Packet Switch testet System auf Fehler und konfiguriert Weiterleitungstabelle automatisch

Dynamisches Routing im WAN (2)

- Darstellung des WAN als Graph und Berechnung kürzester Pfade
  - Knoten sind Packet Switches, Kanten sind Verbindungen

- Effiziente Algorithmen in Graphentheorie bereits vorhanden



WAN mit zugehörigem Graph

to reach	next hop
1	–
2	(1,3)
3	(1,3)
4	(1,3)

Node 1

to reach	next hop
1	(2,3)
2	–
3	(2,3)
4	(2,4)

Node 2

to reach	next hop
1	(3,1)
2	(3,2)
3	–
4	(3,4)

Node 3

to reach	next hop
1	(4,3)
2	(4,2)
3	(4,3)
4	–

Node 4

Weiterleitungstabelle

Default Route

- Switch 1 in letztem Beispiel hat nur eine Verbindung → Next-Hop ist immer Switch 3
- In großem WAN kann es hunderte Duplikate geben
- Default Route:
  - Optionaler Eintrag in Weiterleitungstabelle ersetzt lange Listen mit selben Eintrag
  - Nur eine Default Route pro Tabelle ist erlaubt
  - Genutzt falls Routing Software keinen passenden Eintrag findet

to reach	next hop
1	–
*	(1,3)

Node 1

to reach	next hop
2	–
4	(2,4)
*	(2,3)

Node 2

to reach	next hop
1	(3,1)
2	(3,2)
3	–
4	(3,4)

Node 3

to reach	next hop
2	(4,2)
4	–
*	(4,3)

Node 4

Default Route als Stern gekennzeichnet

Berechnung Weiterleitungstabelle

- Statisches Routing:
  - Programm berechnet und installiert Routen bei Booten des Switch
  - Routen ändern sich nicht
  - Einfach, wenig Overhead
  - Nicht sehr flexibel

- Dynamisches Routing:
  - Programm erstellt initiale Weiterleitungstabelle bei Booten
  - Ändert Tabelle bei Änderungen im Netzwerk
  - In meisten WANs genutzt

Verteilte Routenberechnung

- Dijkstra Algorithmus berechnet kürzesten Pfad in Routen
- Aber: Verteilte Routenberechnung benötigt
- Jeder Packet Switch soll Weiterleitungstabelle lokal berechnen
- Zwei Ansätze
  - Link-State Routing (LSR) nutzt Dijkstra
  - Distance-Vector Routing (DVR)

Link-State Routing (LSR)

- Auch: Shortest Path First (SPF)
- Switches senden periodisch Broadcast mit Status der Verbindungen zwischen zwei Switches
- Software in Switch sammelt Nachrichten und erstellt Graph des Netzwerk
- Nutzt Dijkstra Algorithmus um Weiterleitungstabelle zu berechnen
- Falls Verbindung ausfällt wird dies von beteiligtem Switch bemerkt und als Status verschickt → Switches passen Tabelle an

Dijkstra Algorithmus (1)

- **Gegeben**: Graph mit nichtnegativen Gewichten an jeder Kante und ausgewählte Quelle

- **Berechnen**: Kürzeste Distanz zwischen Quelle und jedem anderen Knoten sowie Weiterleitungstabelle

- **Verfahren**:

  Initialisiere: Menge S mit allen Knoten außer der Quelle

  Initialisiere: Array D mit D[v] als Gewicht der Kante zwischen Quelle und v bei Existenz, sonst Infinity

  Initialisiere: Einträge in R mit R[v] = v falls Kante zwischen Quelle und V, sonst 0

Dijkstra Algorithmus (2)

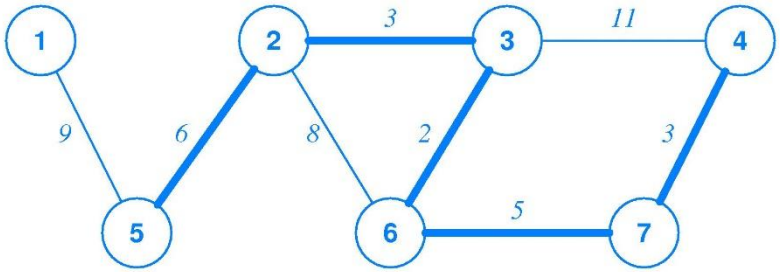
- **Verfahren (Fortsetzung)**

```
while (Menge S nicht leer):
    Wähle Knoten u aus S, so dass D[u] minimal
    if (D[u] = Infinity):
        Error: Kein Pfad zu Knoten in S, Exit
    Lösche u aus Menge S

    for each v mit (u,v) ist Kante:
        if (v in S enthalten):
            c = D[u] + weight(u,v)
            if (c < D[v]):
                R[v] = R[u]
                D[v] = c
```

Dijkstra Algorithmus (3)

- Algorithmus besitzt Menge von Knoten S für die Distanzen und Next-Hops noch nicht berechnet sind
- Betrachtet Element u in S mit kürzester Entfernung und analysiert Verbindungen zwischen u und seinen Nachbarn
- Falls neuer, kürzerer Pfad von Quelle über u zu einem Knoten in S gefunden wird, aktualisiert es Distanz und Next-Hop



Graph mit Gewichten, kürzester Pfad zwischen 4 und 5 gekennzeichnet

Dijkstra Algorithmus (4)

- D und R können Arrays sein, indexiert mit Knoten-ID
  - Eintrag i in D entspricht Distanz von Quelle zu Knoten i
  - Eintrag i in R entspricht Next-Hop für Knoten i
- Menge S kann als doppelt-verkettete Liste gespeichert werden
  - Einfaches Suchen und Löschen
- Infinity kann als Summe aller Pfade + 1 realisiert werden

Gewichte im kürzesten Pfad

- Dijkstra Algorithmus kann andere Gewichte als geographische Entfernung nutzen

- Möglichkeiten
  - Anzahl an Switches im Pfad (jede Kante hat Gewicht 1)
  - Kapazität der Verbindungen
  - Gewichte anhand Policy (Primärer Pfad, Backup Pfad)

DVR (1)

- **Distance-Vector Routing (DVR)**

- Ähnlichkeiten zu LSR
  - Verbindung hat Gewicht, Distanz zu Ziel ist Summe der Gewichte
  - Nachrichten werden periodisch ausgetauscht

- DVR sendet komplette Liste der Ziele und derzeitige Kosten um jedes zu erreichen

- Nachricht mit Paaren (Ziel, Distanz) wird nur an Nachbarn gesendet

DVR (2)

- Bei Empfang einer Nachricht von Nachbarn wird geprüft, ob Nachbar kürzeren Pfad kennt
  - Nachbar N hat Pfad zu Ziel D mit Kosten 5, derzeitiger Pfad zu D über Knoten K hat aber Kosten 100
  - Next Hop für D wird durch N ersetzt, Kosten sind 5 plus die Kosten um N zu erreichen

- In LSR kennen Knoten Problem zur gleichen Zeit (Nur Verzögerung bis Nachricht angekommen ist)

- In DVR berechnet jeder Switch erst Tabelle und sendet dann weiter → Dauert länger bis alle Switches Problem kennen

DVR Algorithmus (1)

- **Gegeben**:
  - Lokale Weiterleitungstabelle mit Distanz pro Eintrag
  - Distanz um Nachbarn zu erreichen
  - Eingehende DV Nachricht

- **Berechnen**: Aktualisierte Weiterleitungstabelle

- **Verfahren**:

  - Initialisiere: Weiterleitungstabelle mit einem einzigen Eintrag (Ziel ist lokaler Switch, Next-Hop leer, Distanz 0)

DVR Algorithmus (2)

Repeat forever:

Warte auf Routing Nachricht von Nachbarn, Sender ist N

for each (Eintrag in Nachricht):

V ist Ziel in Eintrag, D ist Distanz

Berechne C als D plus Gewicht der Verbindung über die Nachricht erhalten wurde

if (Keine Route zu V vorhanden):

Eintrag in Weiterleitungstabelle für Ziel V mit Next-Hop N und Distanz C

else if (Route existiert mit Next-Hop N):

Ersetze Distanz in Route durch C

else if (Route existiert mit Distanz größer C):

Ersetze Next-Hop mit N und Distanz mit C

Routingprobleme (1)

- In Theorie berechnen LSR und DVR kürzeste Pfade korrekt und konvergieren
- Gehen in LSR Nachrichten verloren, haben Switches andere Ansichten über kürzesten Pfad
- In DVR kann Routing Loop entstehen
  - Packet Switch denkt der jeweils andere Switch hat den kürzesten Pfad
  - Paket wird immer hin- und hergeschickt

Routingprobleme (2)

- Backwash in DVR
  - Switch sendet Nachricht: Knoten 1 mit Kosten 3 erreichbar
  - Bei Ausfall der Verbindung entfernt er diesen Eintrag
  - Anderer Switch sendet: Knoten 1 über Kosten 4 erreichbar (basiert auf der falschen Information)

- In Praxis: Beschränkungen und Heuristiken um Schleifen zu vermeiden
  - Switch Horizon in DVR: Information wird nicht an Ursprung der Nachricht gesendet
  - Verhinderung vieler Änderungen in kurzer Zeit

- Dennoch: Probleme in großen Netzwerken, falls oft viele Links ausfallen oder aktiv werden

Zusammenfassung

- WAN kann genutzt werden um Netzwerke über große Distanzen zu bilden und viele Computer zu vernetzen
- WAN besteht ursprünglich aus verbundenen Packet Switches
- Packet Switching verwendet Store-and-Forward Paradigma
- Weiterleitungstabelle besitzt Eintrag mit Ziel und Next-Hop
- Routing Software verwendet Link-State Routing und Distance-Vector Routing

Sonstige Technologien

Zugangstechnologien (1)

- **Synchronous Optical Network Or Digital Hierarchy (SONET/SDH)**
  - Ursprünglich System um digitale Sprachtelefonie zu übertragen
  - Bildet physischen Ring um Redundanz zu gewährleisten
  - Hardware erkennt und korrigiert Fehler
  - Add-Drop Multiplexor um Standort zu SONET Ring zu verbinden: Hinzufügen oder Terminieren von Datenleitungen an Ring
  - Nutzt Time Division Multiplexing

Zugangstechnologien (2)

- **Optical Carrier (OC) Circuits**
  - Spezifizieren Signale in Glasfaser SONET Ring
  - Firma kann OC Circuit mieten um zwei Standorte zu verbinden
  - Tier 1 ISPs nutzen Circuits zwischen OC-192 (10 Gbps) und OC-768 (40 Gbps)

- **Digital Subscriber Line (DSL) und Kabelmodems**
  - Bieten Breitbandanbindung für Privatleute und kleine Firmen
  - DSL nutzt existierende Telefonleitungen, Bandbreite abhängig von Distanz zwischen Kunde und Anbieter
  - Kabeltechnologie nutzt existierende Infrastruktur für Kabelfernsehen, Bandbreite zwischen Nutzern geteilt
  - Technologien als Übergang bis Glasfaser für zu Hause verfügbar

Zugangstechnologien (3)

- **Wi-Fi**
  - Kabellose Technologien
  - Bieten Internetzugang zu Hause, in Flughäfen, Hotels, …
  - Datenrate wird immer weiter gesteigert (Aktuell Geräte für 802.11ac mit 1300 Mbit/s)

- **WiMAX**
  - Kann zur Bildung eines MAN genutzt werden
  - Bietet Zugang sowie Backhaul (Entfernter Ort an Provider anbinden)
  - Feste und mobile Endpunkte

LAN Technologien (1)

- **Token Ring**
  - Zugangskontrolle über ausgetauschten Token
  - IBM Token Ring war weit verbreitet
  - Initial 4 Mbps, später 16 Mbps

- **Fiber And Distributed Data Interconnect (FDDI)**
  - Ende 1980er: Ethernet mit 10 Mbps und IBM Token Ring mit 16 Mbps unzureichend
  - Erhöhte Datenraten in LAN auf 100 Mbps, nutzte Glasfaser
  - Redundanz in FDDI durch gegenläufige Ringe
  - Einer der ersten LAN Switches
  - Physisch Sterntopologie, logisch Ringtopologie
  - Hohe Kosten und Spezialwissen notwendig
  - CDDI: Version von FDDI über Kupferkabel

LAN Technologien (2)

- **Ethernet**
  - Dominiert Markt, öfter eingesetzt als alle anderen Technologien
  - Kaum mehr Ähnlichkeit zwischen frühen Ethernet Versionen (dicke Koaxialkabel) und heutigem Gigabit Ethernet
  - Hubs ersetzten Kabel, Switches ersetzten Hubs, VLAN Switches ersetzen Switches

WAN Technologien (1)

- **ARPANET**
  - Advanced Research Projects Agency (ARPA) förderte Forschungsprojekte für US Department of Defense
  - ARPANET war eines der ersten WANs mit Packet Switching
  - Verbindungen zwischen Universitäten und Industrie
  - Verbindungen nutzten serielle Leitungen mit 56 Kbps
  - Konzepte, Algorithmen, Terminologie oft noch heute genutzt
  - Forscher am Internet Projekt kommunizierten und experimentierten über ARPANET
  - Ab 1983 Verwendung der Internet Protokolle

Voice and Voice Over IP (VoIP)

- Transport von Echtzeitsprache und Video
- IETF entwickelte SIP
- ITU entwickelte H.323

Software Defined Networking

- Trennung des Netzwerkmanagements von der Netzwerkhardware
  - Netzwerkkomponenten lassen sich über eine standardisierte Softwareschnittstelle programmieren / steuern
- Z.B. Openflow
- Wechseln der Hardware oder gar des Herstellers einfach