

# ATM

## Vorlesung

### Rechnernetze und Internet – Fortgeschrittene Themen

20. Juni 2002

SS 2002 – Veranstaltungsnummer 260156

Guido Wessendorf

Zentrum für Informationsverarbeitung

Westfälische Wilhelms-Universität Münster

E-Mail: [wessend@uni-muenster.de](mailto:wessend@uni-muenster.de)



## Motivation für ATM

- neue Technologie für digitale Kommunikation nötig
  - universell
  - global
  - effizient
- Integration *aller* digitalen Kommunikationsformen mit einer Technologie
  - traditionelle Telefonie
  - Audio
  - Video
  - Datentransfer
  - ...



## Historie von ATM

- 1983, 1984: CNET, AT&T Bell Labs und Alcatel Bell beginnen Entwicklung von ATM
- Ende 80'er: CCITT: B-ISDN (Broadband Integrated Services Digital Network)
- 1991: ATM-Forum: Zusammenarbeit von Industrie und Telcos zur rascheren Entwicklung und Standardisierung von ATM (z.Z. über 600 Mitglieder)
- 1993: Phase 1 der (Basis-) ATM-Standardisierung abgeschlossen

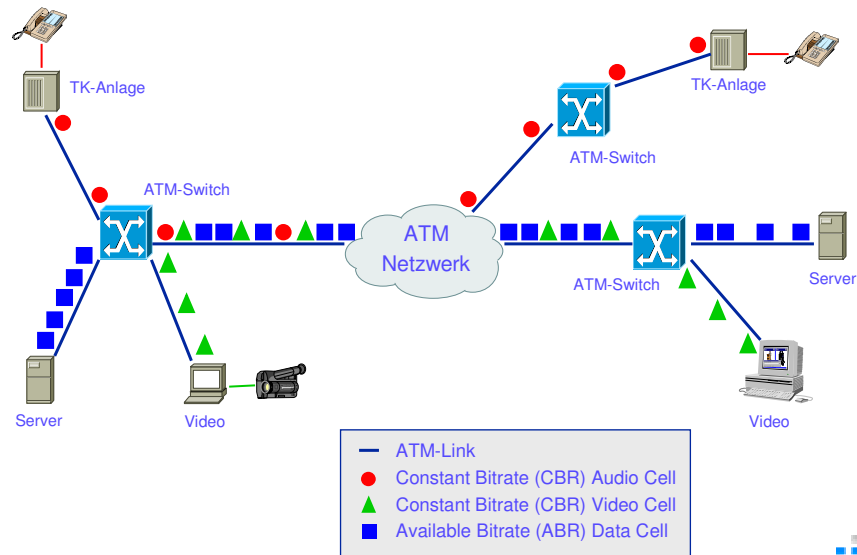


## ATM im Überblick

- Asynchronous Transfer Mode (ATM)
- verbindungsorientierte Technologie
- ATM-Zellen haben feste Länge von 53 Oktetts
- effiziente Switching-Hardware möglich
- Unterstützung verschiedener Dienstklassen
- Konzept unabhängig vom physika. Übertragungsmedium



## Integrierte Dienste mit ATM



ATM, 20. Juni 2002, Guido Wessendorf

5

## ATM-Schichten-Modell

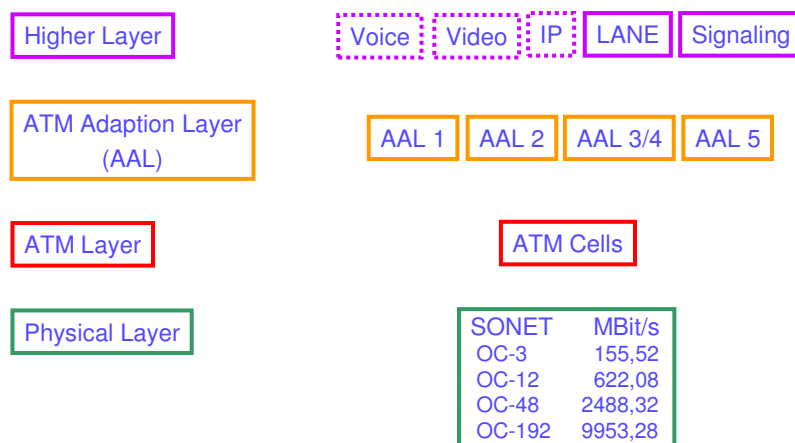


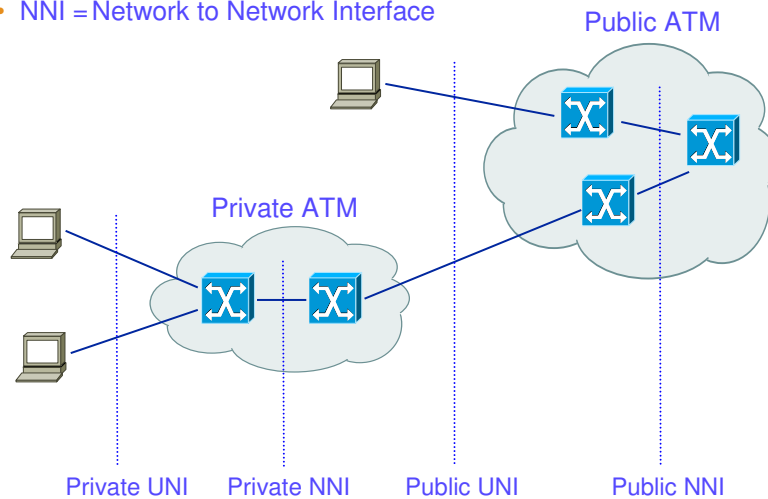
Abbildung nicht vollständig!

ATM, 20. Juni 2002, Guido Wessendorf

6

## ATM-Schnittstellen

- UNI = User to Network Interface
- NNI = Network to Network Interface

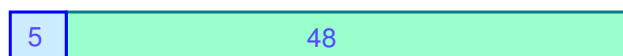


ATM, 20. Juni 2002, Guido Wessendorf

7

## ATM-Zelle

- feste Länge von 53-Oktetts
- 5 Oktett Header, 48 Oktett Daten



UNI-Zelle:

GFC	VPI
VPI	VCI
VCI	
VCI	PTI
HEC	

← 8 Bit →

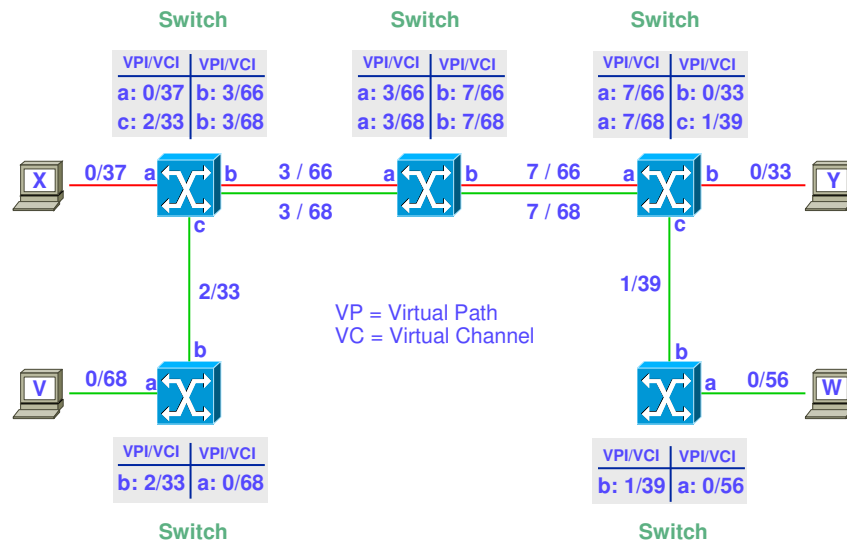
- GFC: Generic Flow Control
- VPI: Virtual Path Identifier
- VCI: Virtual Channel Identifier
- PTI: Payload Type Identifier
- CLP: Cell Loss Priority
- HEC: Header Error Control

- NNI-Zelle enthält kein GFC-Feld. Dafür ist VPI-Feld 12-Bit lang

ATM, 20. Juni 2002, Guido Wessendorf

8

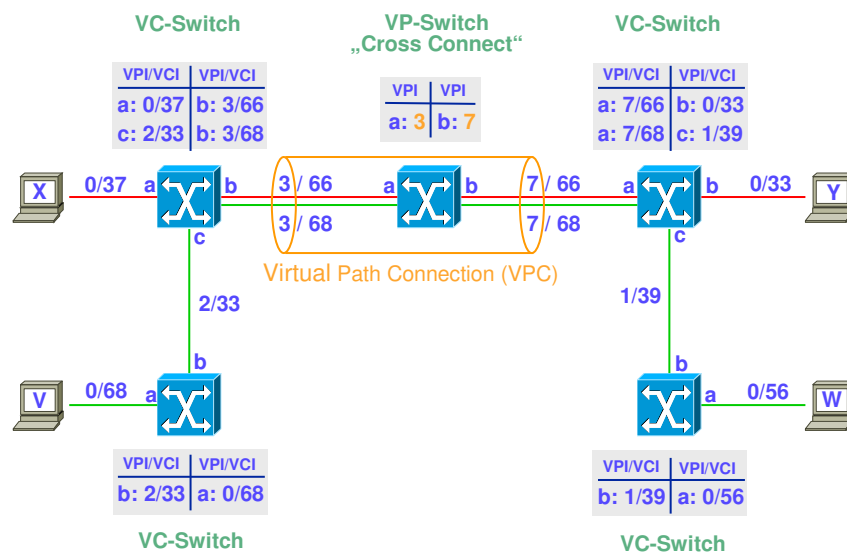
## ATM-Verbindungen



ATM, 20. Juni 2002, Guido Wessendorf

9

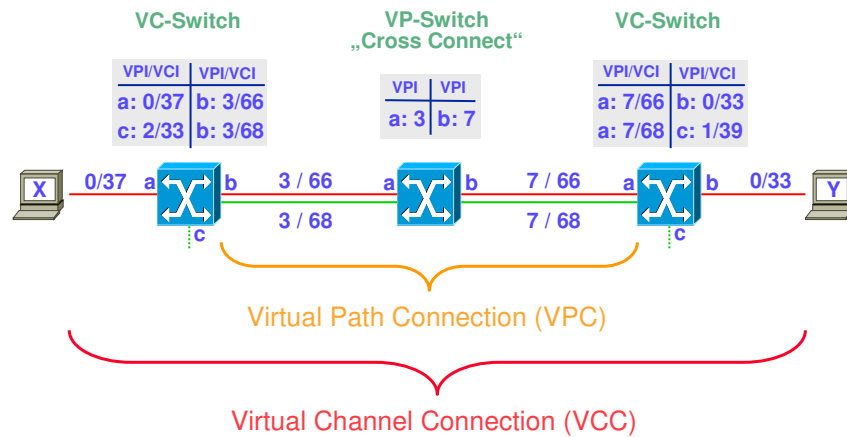
## ATM-Verbindungen (Cross Connect)



ATM, 20. Juni 2002, Guido Wessendorf

10

## Bezeichnungen: VCC und VPC



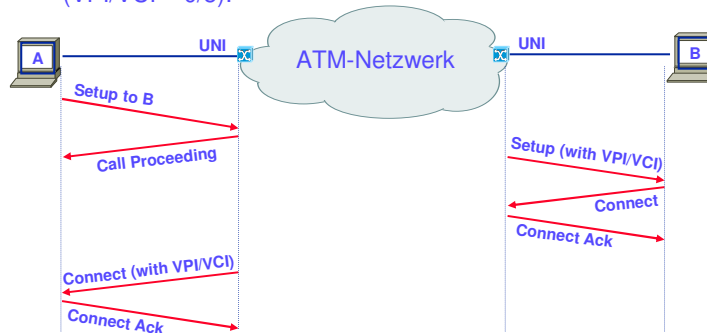
## Verbindungs-Arten

- Permanent Virtual Circuit (PVC)
  - VPI/VCI-Tabellen werden manuell vom Administrator erstellt
  - PVCs bleiben auch bei ausbleibendem Datenverkehr permanent bestehen
- Switched Virtual Circuit (SVC)
  - Verbindungen werden dynamisch mittels ATM-Signaling auf- und wieder abgebaut
  - SVCs haben i.d.R. eine "Verfallszeit"
- Soft PVC
  - Verbindungen werden an UNI-Schnittstellen manuell und an NNI-Schnittstellen automatisch konfiguriert

## ATM-Signalisierung

- da ATM verbindungsorientiert ist, müssen Nachrichten für den Verbindungsaufbau, während einer Verbindung und für den Verbindungsabbau ausgetauscht werden
- Signalisierungsnachrichten enthalten u.a. Verbindungs-Identifikatoren (VPI/VCI), Ziel-ATM-Adresse (NSAP), Zell-Rate, QoS-Klasse, ...

**Beispiel:** Verbindungsaufbau von A nach B an der UNI-Schnittstelle (VPI/VCI = 0/5):



ATM, 20. Juni 2002, Guido Wessendorf

13



## ATM Adressen

- für (globale) Adressierung von ATM-Interfaces
  - ermöglichen (globales) „Routing“ (s.u.) der ATM-Signalisierungs-Zellen und somit einen dynamischen (globalen) Verbindungsaufbau
- OSI NSAP (Network Service Access Point) Format (ISO 8348)

- Data Country Code (DCC) (ISO 3166), AFI = 39



**Beispiel:** 39.276F.3100.0119.0000.0001.0101.CA0001340001.05

- International Code Designator (ICD) (ISO 6523), AFI = 47



- NSAP Format E.164 (⇒ ISDN-Nummern), AFI = 45



DSP = Domain Specific Part  
ESI = End System Identifier

ATM, 20. Juni 2002, Guido Wessendorf

14



## Interim Local Management Interface (ILMI)

- SNMP über AAL 5 an der UNI-Schnittstelle
- Zugriff auf Management Information Base (ATM-UNI MIB)
- Status- und Kontrollinformationen über VC's
- initiale Adreß-Registrierung



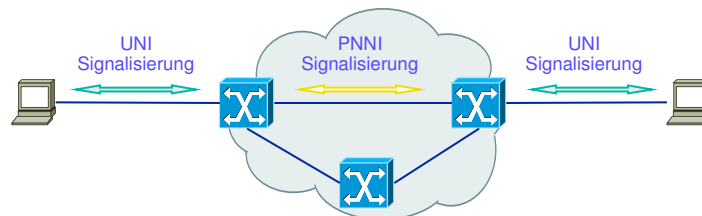
## ATM-Routing

- Interim Interswitch Signaling Protocol (IISP)
  - ATM-Forum, temporäre Lösung
  - statische Routing-Tabellen
  - „PNNI Phase 0“
- Private Network-Network Interface (PNNI) (Ver. 1.0)
  - „PNNI Phase 1“, ATM-Forum, März 1996
  - zwei Komponenten:
    - » PNNI Signalisierungs-Protokoll
    - » PNNI Routing-Protokoll
  - komplexer als jedes andere Routing-Protokoll
    - » QoS-basierte Wegewahl
    - » zugleich für LAN- und WAN-Bereich geeignet
    - » skalierbare hierarchische Netzwerk-Organisation
  - integrierter „Crankback“-Mechanismus



## PNNI-Signalisierung

- PNNI zwischen ATM-Switchen (NNI-Schnittstellen)
- PNNI-Links können physische Links oder virtuelle („multi-hop“) Links sein (→ „Logical Link“)
- ILMI auch für PNNI-Signalisierung über NNI-Links
- PNNI-Signalisierung ist Erweiterung der UNI-Signalisierung um zusätzliche PNNI-spezifische Parameter (z.B. DTL's (s.u.))
  - ebenfalls  $VPI/VCI = 0/5$  (für physische Links, sonst VPI variabel)



ATM, 20. Juni 2002, Guido Wessendorf

17



## PNNI-Routing

- Routen der Signalisierungs-Anforderungen durch das ATM-Netz
  - die gleichen Wege werden anschließend zur Datenübertragung verwendet
- Wegewahl muss QoS-Anforderungen genügen
  - jeder Switch prüft vor einem Verbindungsaufbau, ob geforderte Verkehrs- und QoS-Parameter erfüllt werden können  $\Rightarrow$  CAC (Connection Admission Control), (s.u.)
  - über den gesamten Weg müssen die jeweils lokalen CAC's aller beteiligten Switches den Anforderungen genügen
- PNNI ist Topology State Routing Protokoll
  - ähnlich OSPF (= Link State Routing Protokoll)
  - Switches verteilen PTSE's (PNNI Topology State Elements) untereinander
  - PTSE's enthalten u.a. Erreichbarkeits- und QoS-Infos
  - logischer PNNI-Link kann komplexe Topologie sein

ATM, 20. Juni 2002, Guido Wessendorf

18

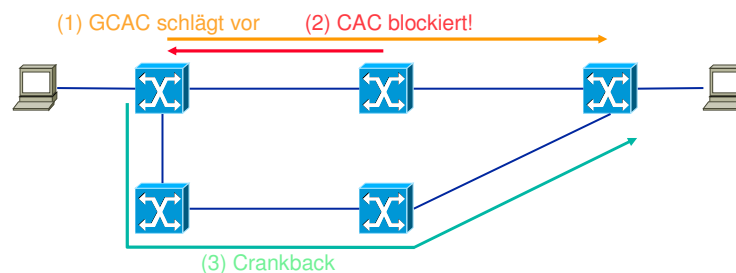


## Source-Routing

- PTSE's aller Switche erlauben Source-Routing (vollständige Wegewahl beim Absender)
  - Vorraberechnung generischer CAC's (GCAC)
  - Switche, welche den Anforderungen nicht genügen, werden aus den möglichen Wegen gestrichen (pruning) und anschließend „kürzester“ Weg berechnet
  - für jedes Ziel wird so eine DTL (Designated Transit List) angelegt
- falls CAC's während Zell-Transit dennoch eine Weiterleitung verbieten (GCAC's evtl. nicht aktuell), wird „Crankback“ durchgeführt

## Crankback

- falls CAC's eine Weiterleitung nicht erlaubt, wird Call bis zu dem Switch zurückgesetzt, der die DTL generiert hat (nicht notwendig der echte Absender, s.u.)
- zugleich werden Infos mitgeliefert, wo der Weg blockiert wurde, und somit eine neue DTL berechnet

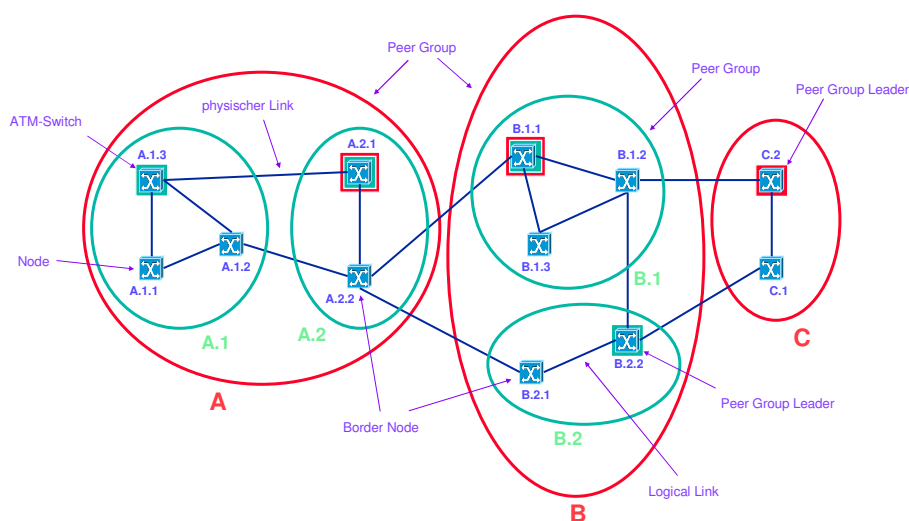


## Hierarchisches Konzept – Terminologie

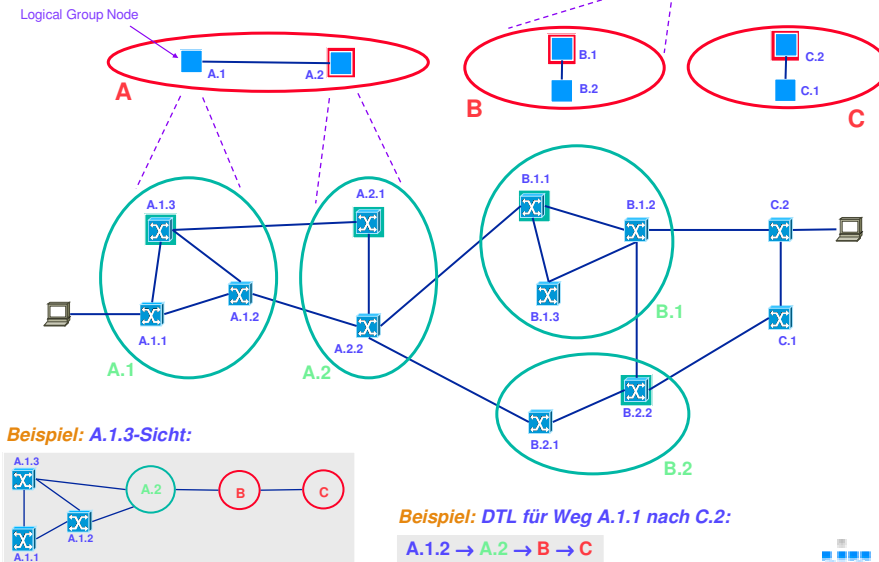
- **Logical Node Group (LGN)**: repräsentiert eine Gruppe von Nodes
- **Logical Node**: Node (Switch) oder LGN
- **Peer Group (PG)**: Gruppe von Logical Nodes in gleicher Hierarchie-Stufe (gleicher längster gemeinsamer Adreß-Präfix = PG-ID)
- **Child Node**: Logical Node auf nächst tieferer Hierarchie-Ebene
- **Parent Node**: LGN auf nächst höherer Hierarchie-Ebene
- **Border Node**: Node mit mind. einem Link zu anderer PG (außerhalb der eigenen Hierarchie, d.h. nicht Child- oder Parent Node)
- **Peer Group Leader (PGL)**: repräsentiert Gruppe auf nächst höherer Hierarchie-Ebene (dort als LGN) – wird automatisch aus allen Nodes einer PG ausgewählt. Ein PGL übernimmt ATM-Adresse vom Switch, jedoch mit anderem SEL



## Hierarchisches-Konzept – Beispiel



## Hierarchische Sicht



## Hierarchie-Bildung

- bei Initialisierung tauschen Nachbar-Nodes PNNI-Hello-Pakete aus (VPI/VCI = 0/18)
- nur wenn Nachbarn in gleicher PG sind werden anschließend PTSE's ausgetauscht (Border Nodes tauschen keine PTSE's aus)
- nach Synchronisation der Datenbasis werden Links propagiert indem PTSE's an *alle* Nodes der eigenen PG verteilt werden (Flooding)
  - jeder Node bestätigt (ack) den Empfang einer PTSE zum jeweiligen Nachbar-Sender, aktualisiert mit den enthaltenen Infos die eigene DB und verteilt dann die PTSE an *alle anderen* Nachbarn weiter
- PTSE's haben Lifetime und werden nach Verfall gelöscht
- PTSE's werden regelmäßig bzw. bei bestimmten Ereignissen (wie z.B. Änderung einer Metrik durch Bandbreitenerhöhung) verteilt
- PGL geben eine zusammenfassende PTSE über die eigene PG an den Parent Node weiter
- nur der PGL darf PTSE's eine Ebene höher verteilen: der PGL ist in der höheren Ebene LGN und tauscht dort PTSE's mit anderen Logical Nodes der gleichen Ebene aus
- Umgekehrt verteilt PGL die zusammengefassten Infos vom Parent Node in der eigenen Ebene

## ATM Adaption Layer (AAL)

- Teil des ATM-Standards
- Bindeglied zwischen den zu übertragenden Nutzdaten und der zellorientierten Übertragung durch die ATM-Schicht (siehe Schichtenmodell)
- regelt die Anforderungen der vier Serviceklassen (A bis D), welche wiederum den vier AAL-Typen (AAL1, AAL2, AAL3/4 und AAL5) zugeordnet sind
- AAL-Layer ist in zwei weitere Sublayer unterteilt:
  - Convergence Sublayer (CS)
    - » verwaltet Zuordnung AAL/QoS zu jeweiligen Zellströmen
  - Segmentation and Reassembly (SAR)
    - » führt Umsetzung ATM-Zelle ↔ Daten-Paket durch



## ATM-Serviceklassen (ITU-T)

Service-Klasse	A	B	C	D
Timing	benötigt		nicht benötigt	
Bitrate	konstant	variabel		
Verbindungs-Mode	verbindungsorientiert			verbindungslos
AAL-Type	1	2	3 / 4 5	3 / 4 5



## AAL-Typen

- AAL 1
  - Transport zeitkritischer Anwendungen mit konstanter Bitrate (z.B. Sprache/Video) oder Emulation von PDH-Pfaden (z.B. E1 oder DS1)
- AAL 2
  - Transport zeitkritischer Anwendungen mit variabler Bitrate (z.B. komprimierte Audi/Video-Daten, Mobilfunk)
- AAL 3/4
  - Anpassung verbindungsorientierter (AAL 3) und verbindungsloser (AAL 4) Datenübertragung (LAN) an ATM-Zellenformat. Enthalten sind zusätzlich Infos aus höheren Schichten (z.B. zur Fehlersicherung, Flusssteuerung und Segment-Kennzeichnungen)
- AAL 5
  - abgespeckte Version von AAL 3/4. Speziell für Frame Relay und TCP/IP optimiert



## ATM-Layer Dienstekategorien (ATM-Forum)

	mögliches Verkehrs-Profil	mögliche Applikation(en)
CBR (Constant Bit Rate)		Sprache, Video
rt-VBR (realtime Variable Bit Rate)		komprimiertes Video/Audio
nrt-VBR (non realtime Variable Bit Rate)		File-Transfer
ABR (Available Bit Rate)		LAN-Datenverbindungen
UBR (Unspecified Bit Rate)		öffentliche WAN-Verbindungen



## Traffic-Contract

- Dienstekategorien (z.B. CBR, VBR, ABR) müssen vor Kommunikationsbeginn beim Verbindungsaufbau „ausgehandelt“ werden
- Zwei Parameterklassen
  - Verkehrs-Parameter
  - Quality-of-Service (QoS)-Parameter



## Parameterklassen

- Verkehrs-Parameter
  - Peak Cell Rate (PCR): Spitzen-Zellrate; maximale Bitrate, die von der Quelle gesendet werden darf
  - Cell Delay Variation Tolerance (CDVT): Toleranz der Zell-Laufzeitschwankungen, bezogen auf Spitzen-Zellrate
  - Sustainable Cell Rate (SCR): Dauer-Zellrate; Obergrenze der mittleren Zellrate, die von der Quelle gesendet werden darf
  - Burst Tolerance (BT): maximale Zellenzahl/Zeit, mit der die Quelle in PCR senden darf
  - Minimum Cell Rate (MCR): minimale Zellrate, die vom Netz garantiert wird
- QoS-Parameter
  - maximum/mean Cell Transfer Delay (max-CTD / mean-CTD): wie lange darf das Netzwerk von Ende-zu-Ende für die Zell-Auslieferung benötigen
  - peak-to-peak Cell Delay Variation (CDV): Varianz im CTD („Jitter“)
  - Cell Loss Ratio (CLR): akzeptierbare Zell-Verlustrate (in %) durch Verwerfen von Zellen bei „Stau“



## Dienstekategorien ↔ Parameterklassen

		CBR	rt-VBR	nrt-VBR	ABR	UBR
QoS-Parameter	CLR	spezifiziert				nicht spezifiziert
	CTD und CDV	CDV und max-CTD		mean-CTD	nicht spezifiziert	
Verkehrs-Parameter	PCR und CDVT	spezifiziert				
	SCR und BT	nicht anwendbar	spezifiziert		nicht anwendbar	
	MCR	nicht anwendbar			spezifiziert	nicht anwendbar

## Traffic-Management

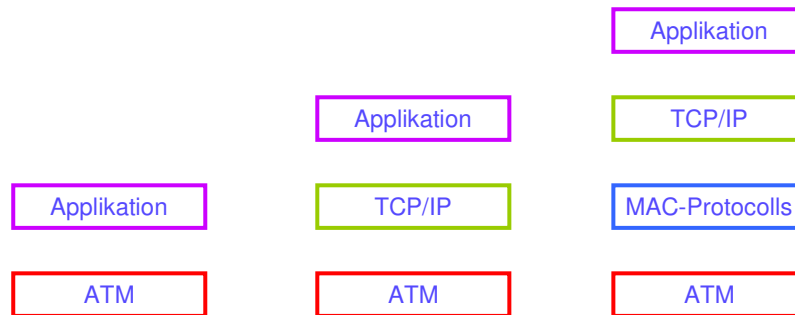
- Connection Admission Control (CAC)
  - bei Verbindungsaufbau-Signalisierung wird Verkehrs-Vertrag bzgl. Verkehrs- und QoS-Parameter überprüft
- Usage Parameter Control (UPC, „Policing“)
  - Überwachung der Verkehrs- und QoS-Parameter (nichtkonforme Zellen werden mit CLP=1 markiert)
- Traffic Shaping
  - Versuch, die zu sendenden Zellen vertragskonform zu halten
  - dynamische Anpassung von Verkehrs-Charakteristika
  - „Leaky Bucket“-Algorithmus



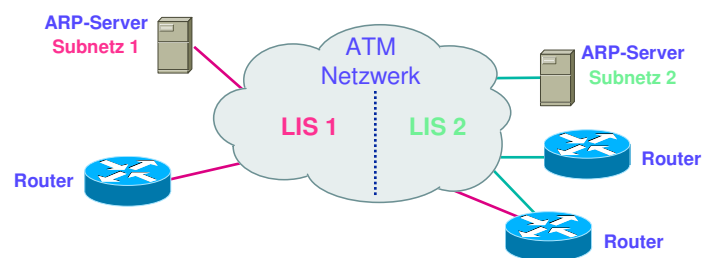


## Einsatzmöglichkeiten von ATM

- Native ATM
- Classical IP over ATM
- LANE
- MPOA



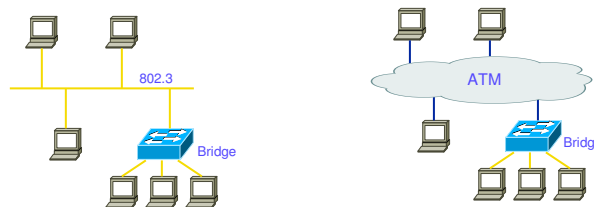
## Classical IP and ARP over ATM (RFC 1577)



- LIS (Logical IP Subnet)
- ARP-Server bildet IP-Adressen auf ATM-Adressen ab
- pro LIS ein ARP-Server
- Router können bei Verkehr zwischen LIS'en nicht umgangen werden
- kein Multicast (⇒ Multicast Address Resolution Server (MARS))

## LAN Emulation (LANE)

- wie Name andeutet: LANE emuliert LAN-Funktionalitäten über ATM-Netze hinweg
  - bietet MAC-Layer Funktionalität wie in Legacy LANs
  - völlig transparent für Applikationen und Protokolle auf höheren Schichten
  - beliebige ans ATM-Netz angebundene Komponenten können über LANE ein gemeinsames ELAN bilden
    - » Bildung virtueller Layer-2 Netze

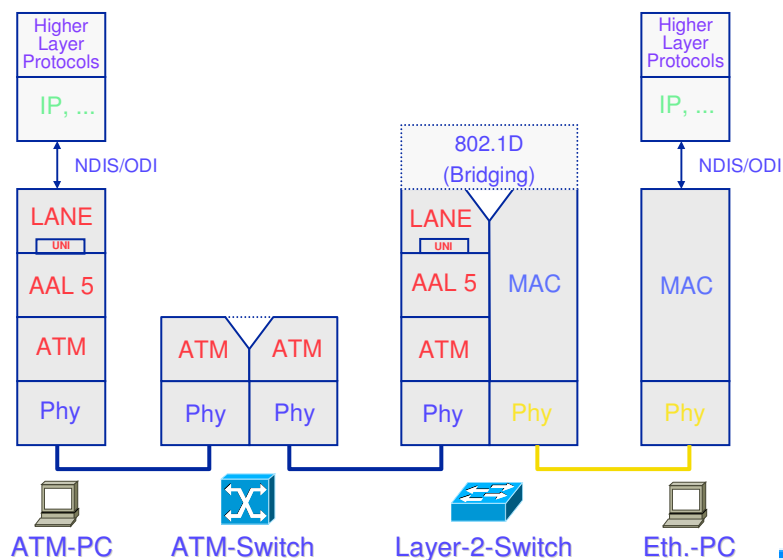


ATM, 20. Juni 2002, Guido Wessendorf

35



## LANE Protokoll Architektur



ATM, 20. Juni 2002, Guido Wessendorf

36



- LAN Emulation Client (LEC)
  - regelt die Teilnahme für ein einzelnes Endsystem (PC, Workstation, Bridge, Router, etc.) an genau einem ELAN  
(mehrere LEC's pro Endsystem für Teilnahme an mehreren ELANs möglich)
  - bietet Standard-LAN-Interface für höhere Layer
  - jeder LEC hat eindeutige ATM-Adresse und wird mit einer oder mehreren MAC-Adressen assoziiert
- LAN Emulation Server (LES)
  - genau ein (aktiver) LES pro ELAN
  - bietet MAC↔LEC-ATM-Address-Auflösung (Anfragen via LE\_ARP - Kommando) und ELAN Kontrollfunktionen
  - jeder LES hat eindeutige ATM-Adresse
  - LEC's melden sich beim LES zur Teilnahme am jeweiligen ELAN an
- Broadcast and Unknown Server (BUS)
  - Multicast-Server für ein ELAN um Verkehr für Ziele mit (noch) unbekannter ATM-Zieladresse und Multicast-Verkehr weiterzuleiten
- LAN Emulation Configuration Server (LECS)
  - Zusatzdienst: löst „lesbare“ ELAN-Namen in ATM-Adressen des jeweils zugehörigen LES auf
  - i.d.R. ein LECS pro ATM-Netz mit „wohlbekannter“ ATM-Adresse oder  $VPI/VCI = 0/17$



## Beispiel-Ablauf

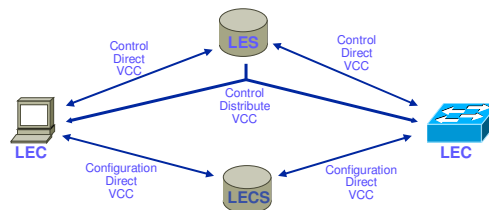
PC-A wird eingeschaltet und soll mit PC-B kommunizieren

1. Initialisierung vom LEC-A
  1. LEC-A führt via ILMI initiale Adreß-Registrierung an S1 durch
  2. LEC-A baut „Configuration-Direct“-VCC zum LECS auf (über wohlbekannte ATM- bzw. VPI/VCI-Adresse) und fragt nach ATM-Adresse vom LES-„Physik“
  3. LECS antwortet mit ATM-Adresse vom LES-„Physik“
2. Join und Registrierung vom LEC-A beim LES-„Physik“
  1. LEC-A baut „Control-Direct“-VCC zum LES-„Physik“ auf
  2. LEC-A registriert eigene ATM- und MAC-Adresse beim LES-„Physik“
  3. LES-„Physik“ baut „Control-Distribute“-VCC zu LEC-A auf (Multicast-Rückkanal)
  4. LEC-A sendet LE\_ARP für BUS-„Physik“-ATM-Adresse zum LES-„Physik“ (via „Control-Direct“-VCC)
  5. LEC-A baut „Multicast-Send“-VCC zum BUS-„Physik“ auf
  6. BUS-„Physik“ baut „Multicast-Forward“-VCC zu A auf (Multicast-Rückkanal)
3. Datentransfer PC-A→PC-B
  1. LEC-A benötigt MAC-Adresse von LEC-B: LE\_ARP-Anfrage an LES-„Physik“ (via „Control-Direct“-VCC)
  2. solange noch keine Antwort vom LES vorliegt, werden alle Pakete zum BUS geleitet (via „Multicast-Send“-VCC)
  3. BUS forwarded Pakete an alle LEC's (via „Multicast-Forward“-VCC) oder direct zu LEC-B (via „Multicast-Send“-VCC zu LEC-B, falls verfügbar)
  4. LEC-A empfängt LE\_ARP-Antwort und baut „Data-Direct“-VCC zu LEC-B auf
  5. A stoppt Pakete Richtung BUS und sendet „LANE-Flush“ um Auslieferung via BUS kontrolliert zu beenden
  6. A sendet Daten weiter, jetzt über den „Data-Direct“-VCC zum LEC-B



## LANE Kontroll- und Daten-VCC's

### • Kontroll-Verbindungen



### • Daten-Verbindungen

