

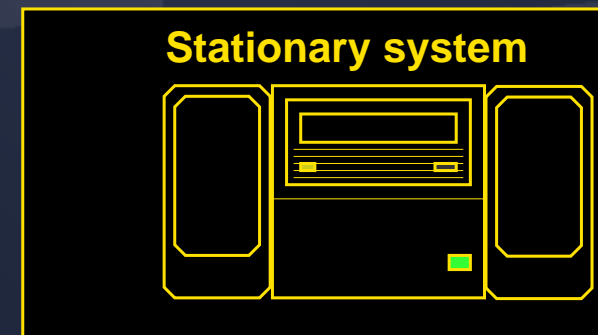
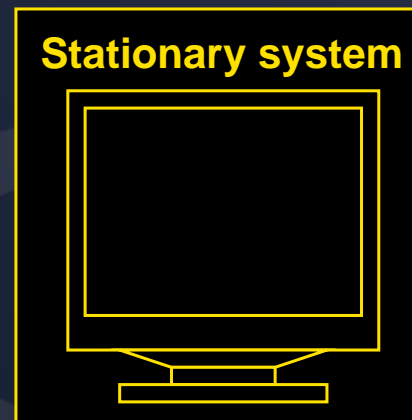
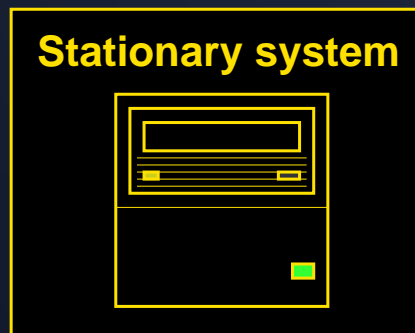
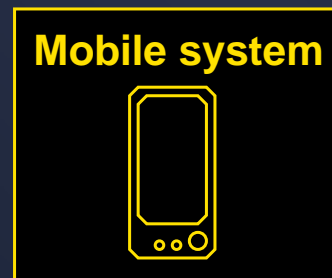
# Towards Automatic Setup of Distributed Multimedia Applications

Marco Lohse and Philipp Slusallek

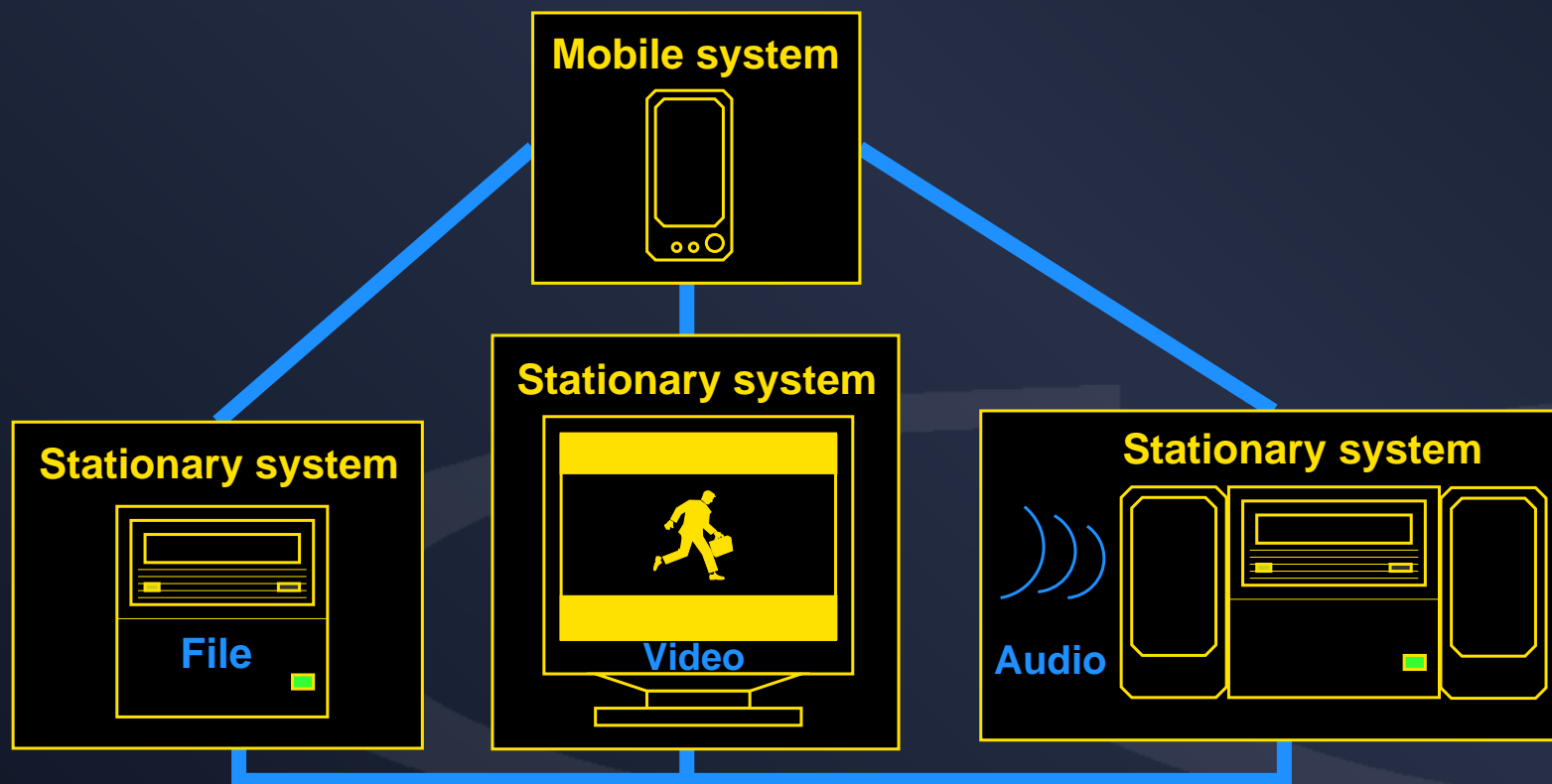
Computer Graphics Lab, Saarland University, Germany

- Growing number of networked multimedia devices
  - PCs, set-top boxes, hi-fi systems, laptop computers, PDAs, mobile phones, ...
  - Networked multimedia home entertainment
- Multimedia middleware solutions
  - Development of distributed multimedia applications
  - Network-Integrated Multimedia Middleware (NMM)

- “Follow-me” application
  - Use capabilities of surrounding environment
  - Remote media sources, processing systems, and rendering devices



- “Follow-me” application
  - Use capabilities of surrounding environment
  - Remote media sources, processing systems, and rendering devices



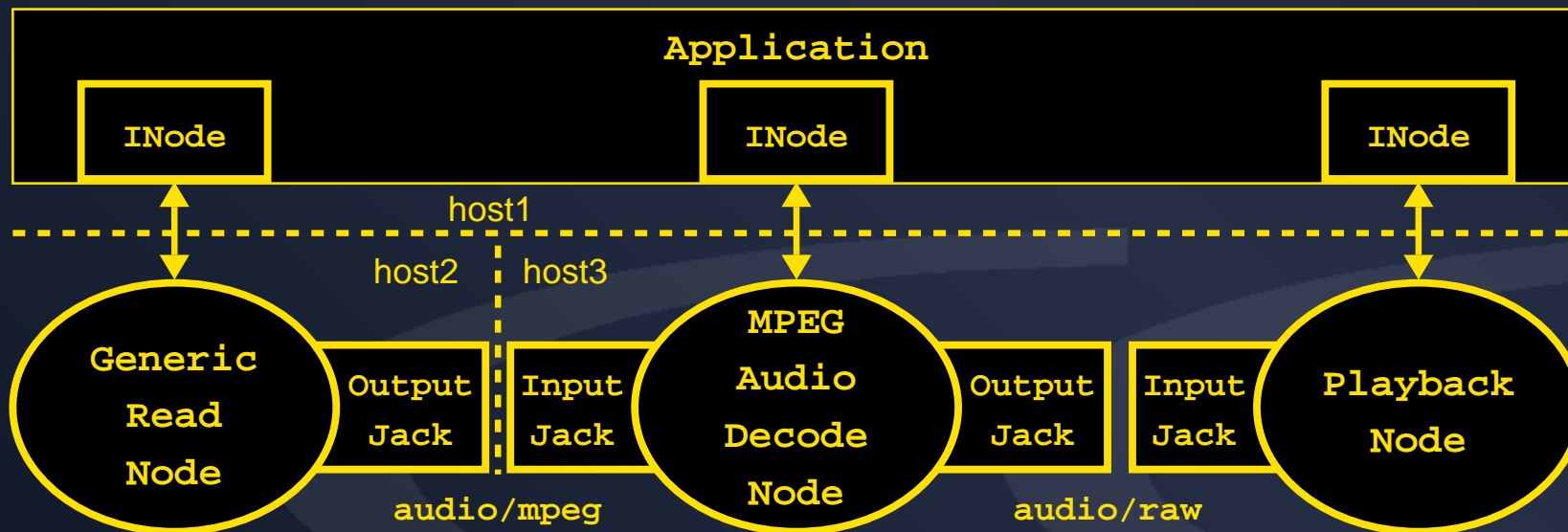
- Locating and integrating distributed components?
- Required format conversions?
- Optimal distribution?

⇒ Too complex to be handled by application developer

- Automatic Application Setup (AAS)
  - Given a high-level description of an intended task plus additional constraints, try to find a valid configuration that meets the given constraints and is distributed in an optimal way

- Motivation
- Underlying multimedia middleware
- Algorithm for Automatic Application Setup (AAS)
- Results
- Conclusions

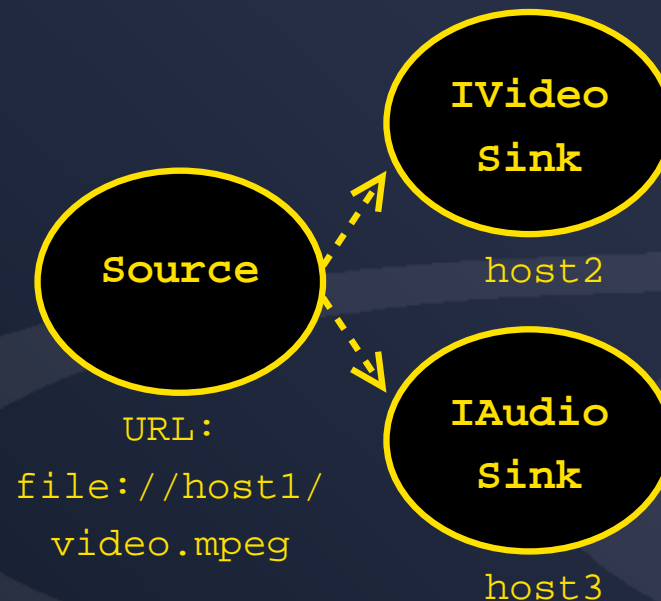
- Distributed flow graphs
  - Local and remote nodes can be controlled and connected
- Node types
  - Source, sink, converter, demultiplexer, ...



- Formats
  - Type/subtype
  - Parameters: Key plus set of values/ranges/wildcard
  - Example: “audio/raw, 44 kHz, 2 channels”
- Registry service
  - Administrates local and remote nodes
  - Query: Name, type, interfaces, supported input/output formats
- Distributed synchronization
  - Inter-stream synchronization of audio/video streams

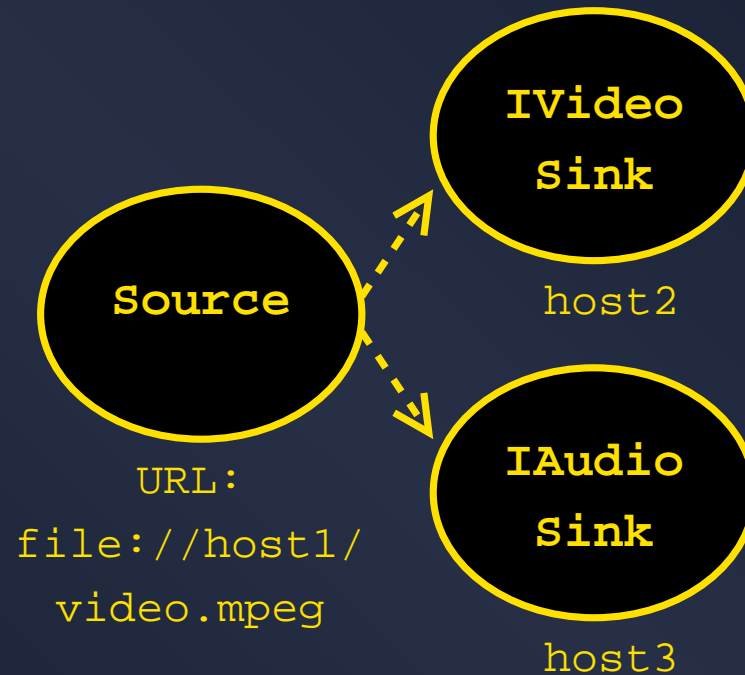
# Algorithm for Automatic Application Setup (AAS)

- High-level specification of intended task
- Distributed media playback
  - URL of source
  - Abstract representation of sinks
  - Locations of all components

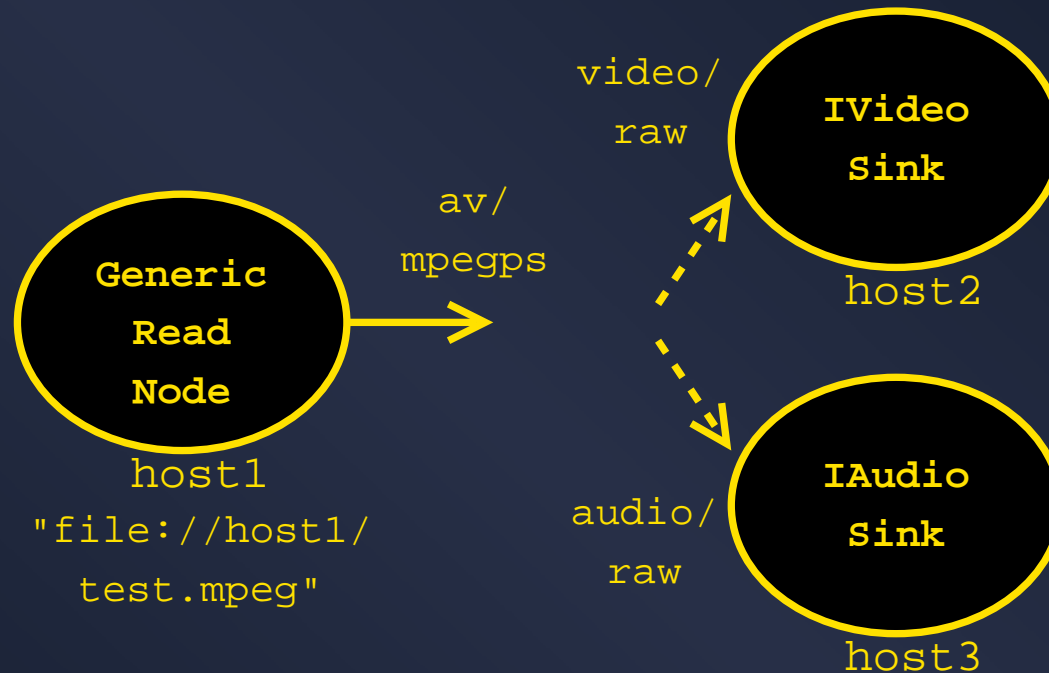


- Find and insert additional intermediate nodes
  - Connection of incompatible formats
- Optimally distribute nodes
  - Minimize network bandwidth
- Iterative step-by-step approach
  - Output formats? Nodes need to process data!
  - “Wrong” choices require backtracking

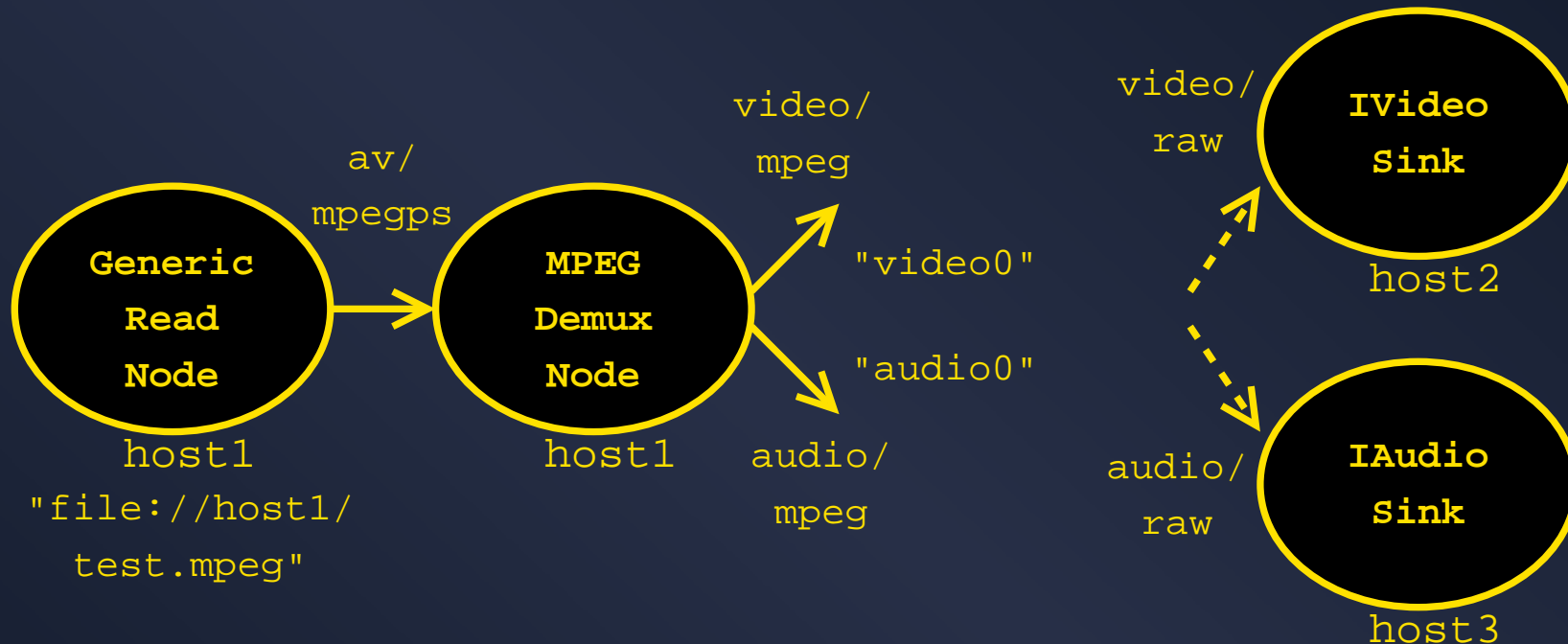
# Graph Builder (0)



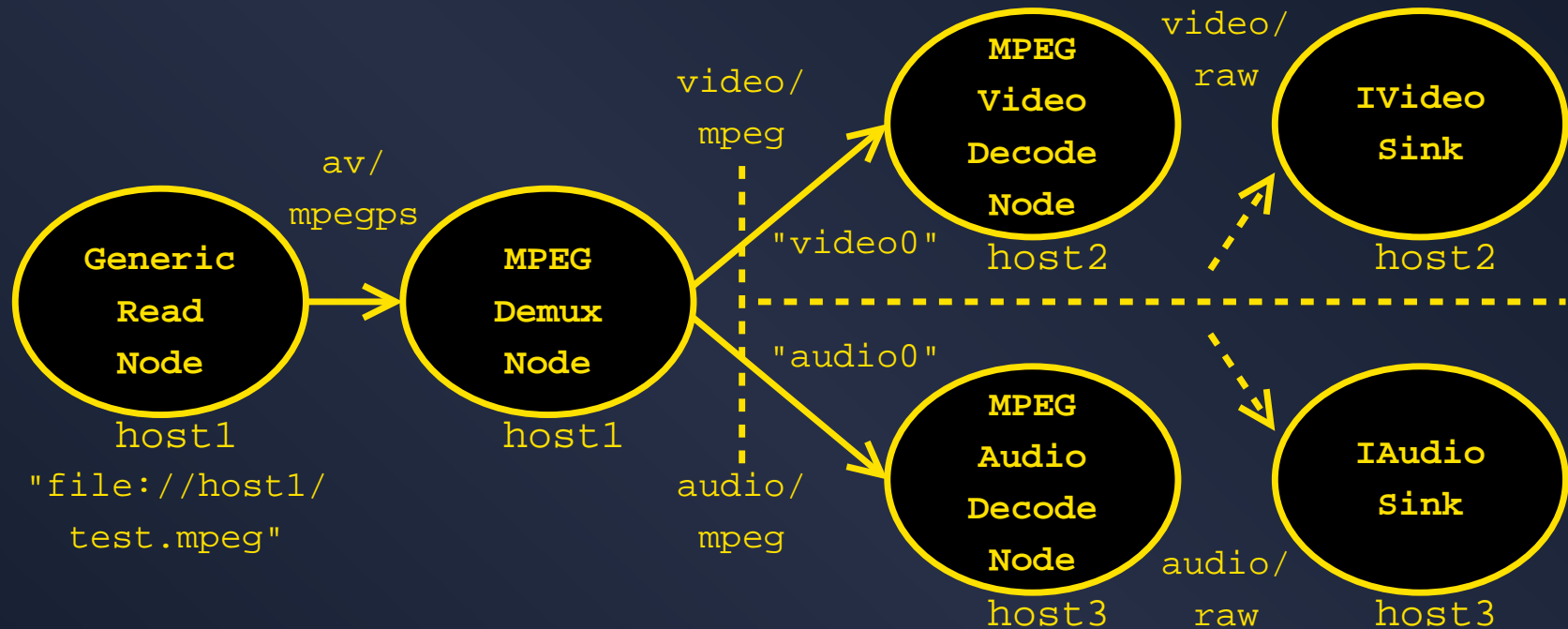
- User graph



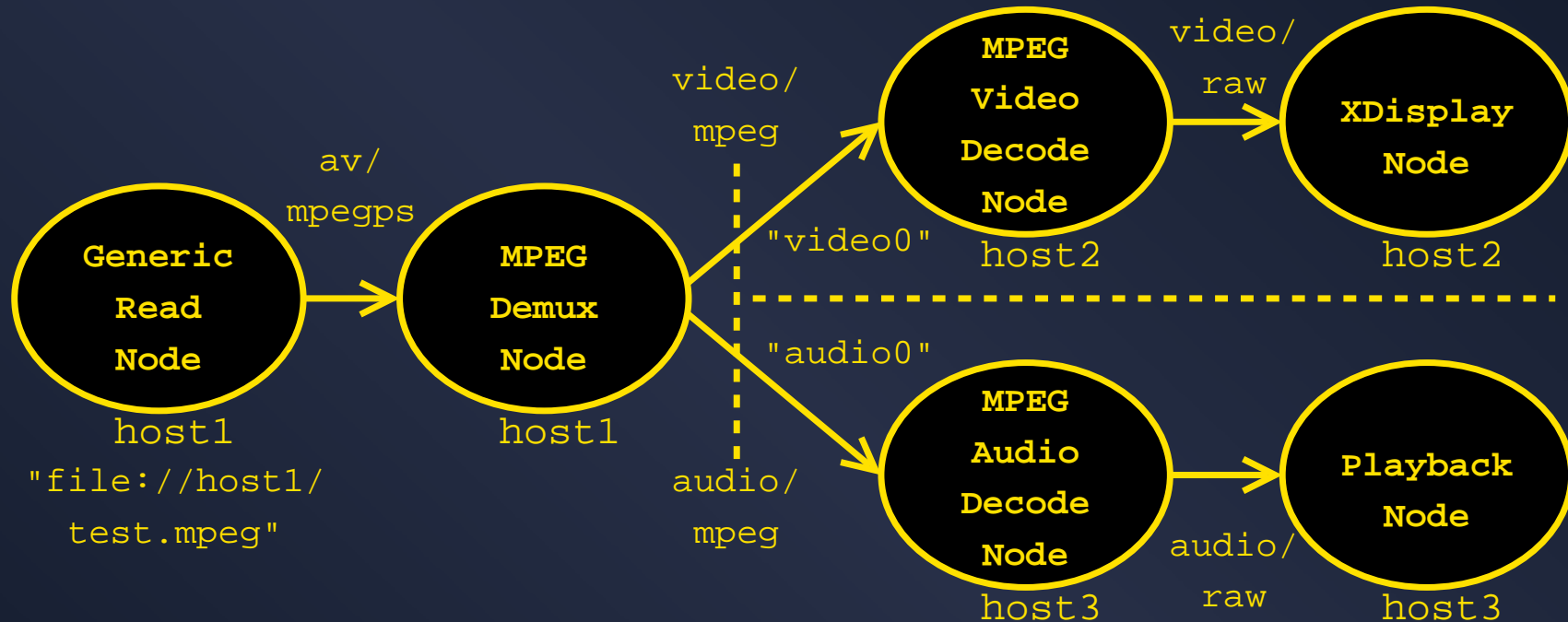
- Determine query for media source from URL
- Determine output format of “left” node by processing data
- Determine available input formats of “right” nodes



- Try to connect current “left” node to “right” nodes
- If not possible, query “matching” intermediate node
  - Converter
  - Demultiplexer

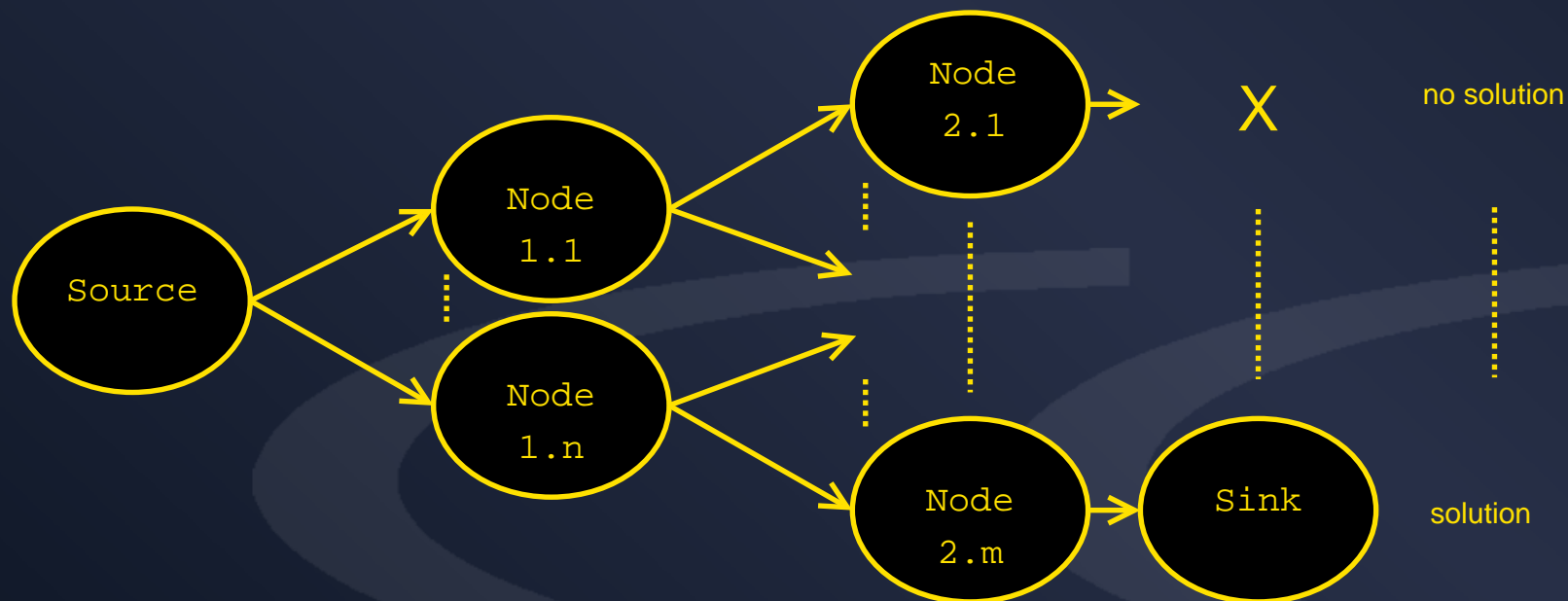


- Optimally distribute intermediate nodes
  - Minimize bandwidth
  - Compare estimated incoming and outgoing bandwidth



- Set up distributed synchronization

- Iterative process  $\Rightarrow$  Backtracking, or ...
- Evaluate branches in parallel
  - Forward data streams to all possible solutions
  - Requires additional resources only during setup
  - Reduces setup delay



Setup					Setup Time (seconds)	
App	File	Audio	Video	Net	Manual	Auto
PC1	PC1	PC1	PC1	-	<b>0.53</b>	<b>0.57</b>
Laptop	Laptop	Laptop	Laptop	-	<b>1.40</b>	<b>1.55</b>
Laptop	PC3	PC1	PC2	LAN	<b>1.50</b>	<b>1.52</b>
Laptop	PC3	PC1	PC2	WLAN	<b>3.10</b>	<b>3.54</b>

- PC1, PC2, PC3: Pentium 4, 3 GHz
- Laptop: Pentium III, 700 MHz
- Network: 100 MBit LAN or 11 MBit WLAN

- Automatic Application Setup
- High-level abstraction
  - User graph
- Graph building algorithm
  - Determines and distributes intermediate nodes
- Applications
  - “Follow-me” media playback, transcoding, streaming server, ...
- Remains challenging problem
  - QoS: CPU resources, network bandwidth, ...



# Questions?