Advances in Methods of Information and Communication Technology, 2014

Modeling Data Flows Hierarchy of an Enterprise Network's ICT Infrastructure

Aleksandr Kolosov <akolosov@cs.petrsu.ru>

Department of Computer Science
Petrozavodsk State University

October 22, 2014
Petrozavodsk, Russia

Current State of the Network Management

- Modern networks are constantly growing.
  - ↑ Services
  - ↑ Complexity
  - ↑ Cost of failures
- Network Management
  - ↓ Risks (downtime, poor quality, security threats)
  - ↓ Cost of ownership
- Problems of the modern network management as defined in the Future Internet Design Initiative report:
  - lack of information of network status and health;
  - a deluge of data;
  - unpredictable effect of control actions.

«. . . a future Internet requires deeply ambitious research in network management.»

— Vint Cerf et al.

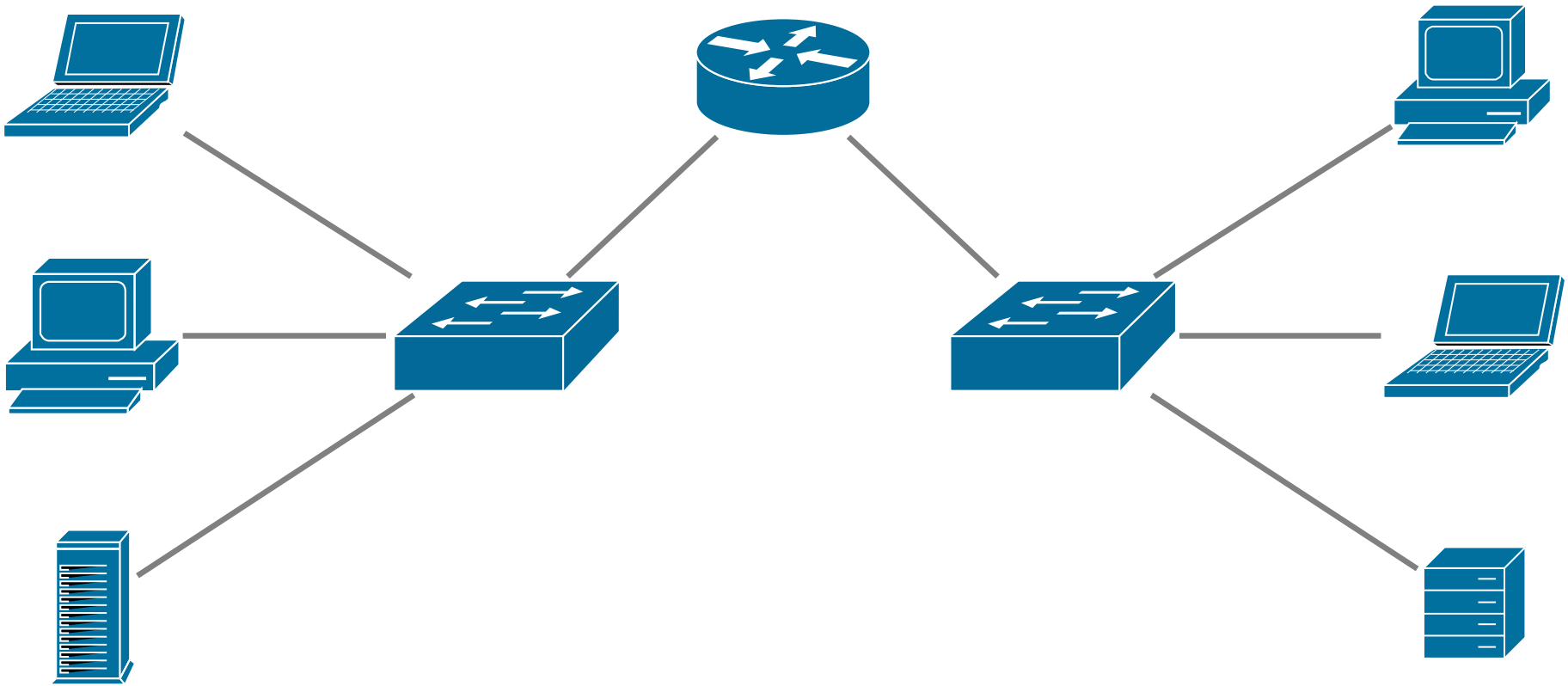
Enterprise Network Specific

- Organizational and spatial structures of the enterprise itself affect traffic patterns and ICT-infrastructure management scenarios.
- Many network management tasks become personnel aware.
- Internal traffic specifics:
  - storage area networks, network attached storages;
  - teleconferencing;
  - virtual workplaces.
- VLANs and VPNs.
- All layers of the network is under consideration when network management tasks are performed.

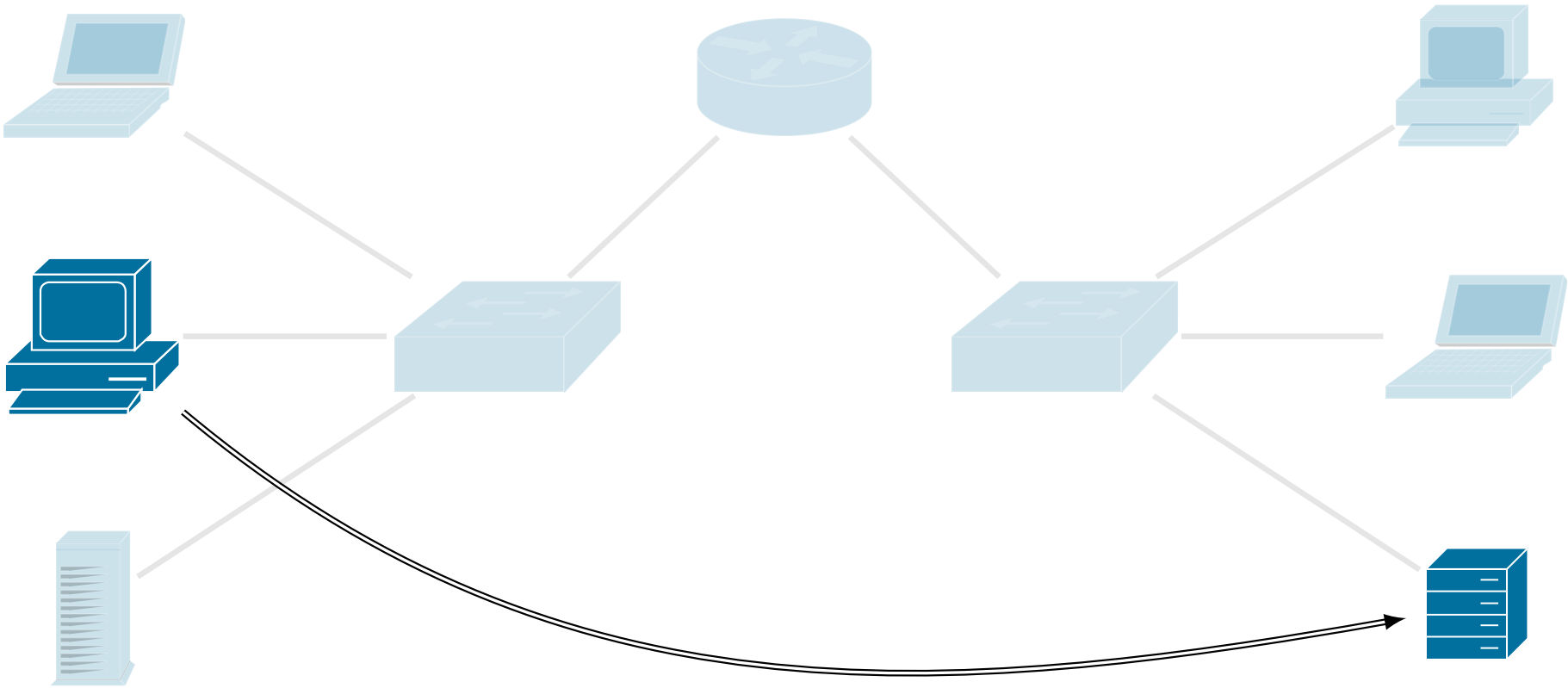
Frontiers of network management

- Research challenges of network management:
  - virtual network environments;
  - maintaining consistency of network state;
  - management friendly protocols and data-plane primitives;
  - scientific methods available for studying network management problems and for evaluating solutions.
- Testbeds should be used to verify new approaches, models and methods of network management:
  - ICT-infrastructure model (network graph + forwarding policies);
  - enterprise structure (spatial and organizational graphs);
  - real traffic data;
  - means for experimental evaluations ("what if" scenarios).
- We are developing such a virtual testbed within the Nest project.

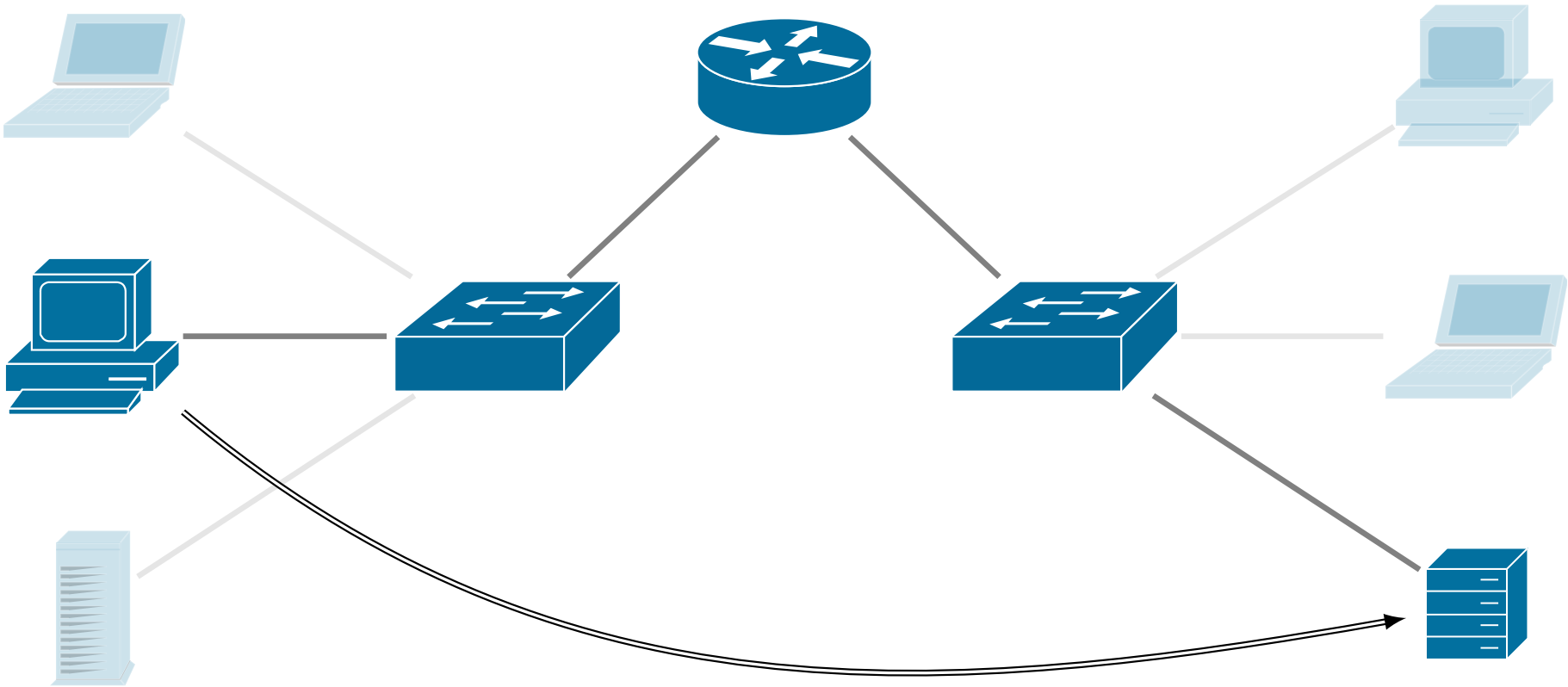
Network Management Methods Virtual Testbed



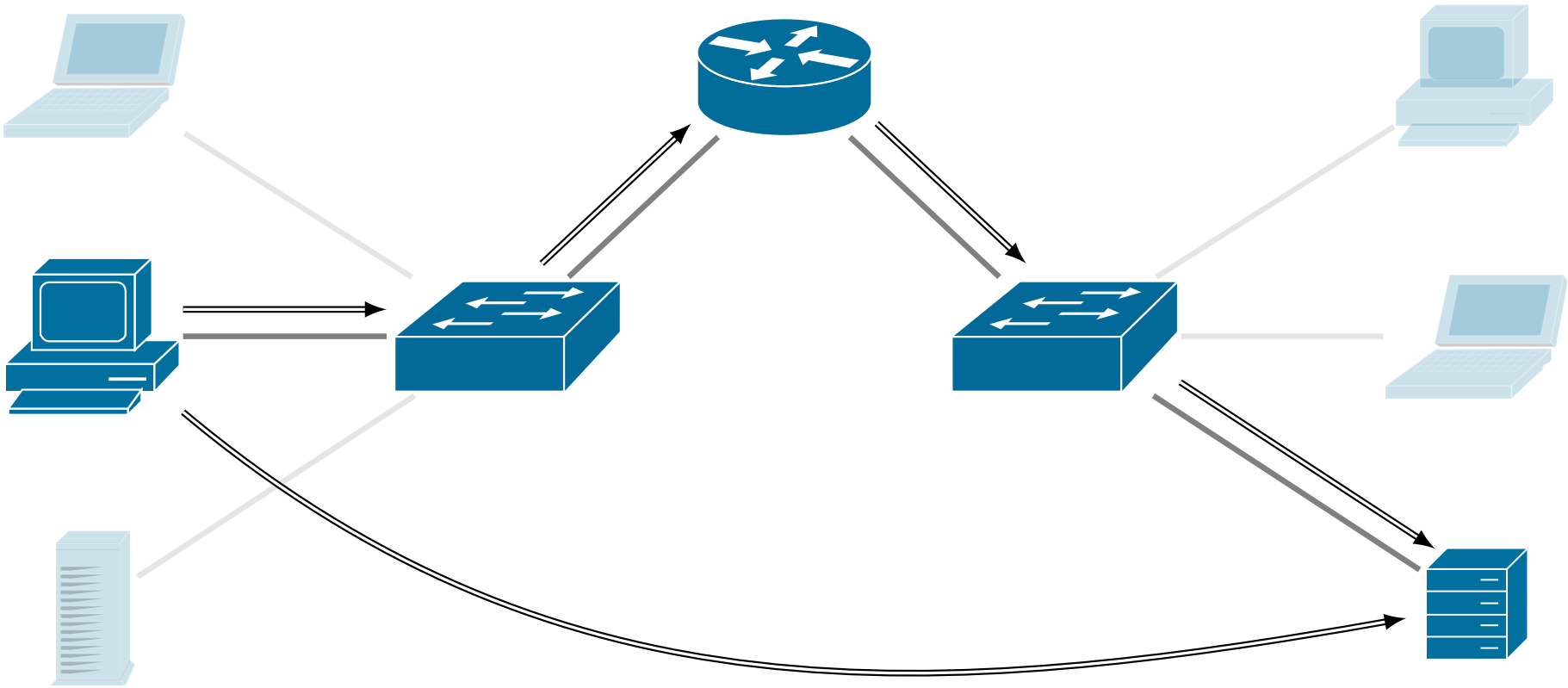
Network Management Methods Virtual Testbed



Network Management Methods Virtual Testbed



Network Management Methods Virtual Testbed



Traffic Measurements Structurization

- We have to map traffic flows to the enterprise architecture graph.
- User must have an opportunity to query any kind of traffic flow in terms of enterprise architecture graph.
- All we need is to enrich a traffic flow concept with the hierarchy.
- There isn't a data source, which produce such measurements:
  - Raw packet data
  - NetFlow / IPFIX
  - Application logs
  - Link statistics
- Taken separately neither source provides required level of details:
  - describe only two points of connection
  - we can not say, is that flow a part of some higher-level communication
  - we can not even say how different records of the source are interconnected
- Analysing all of available data sources together is the key to solving the problem.

Demands for a Traffic Characterization Model

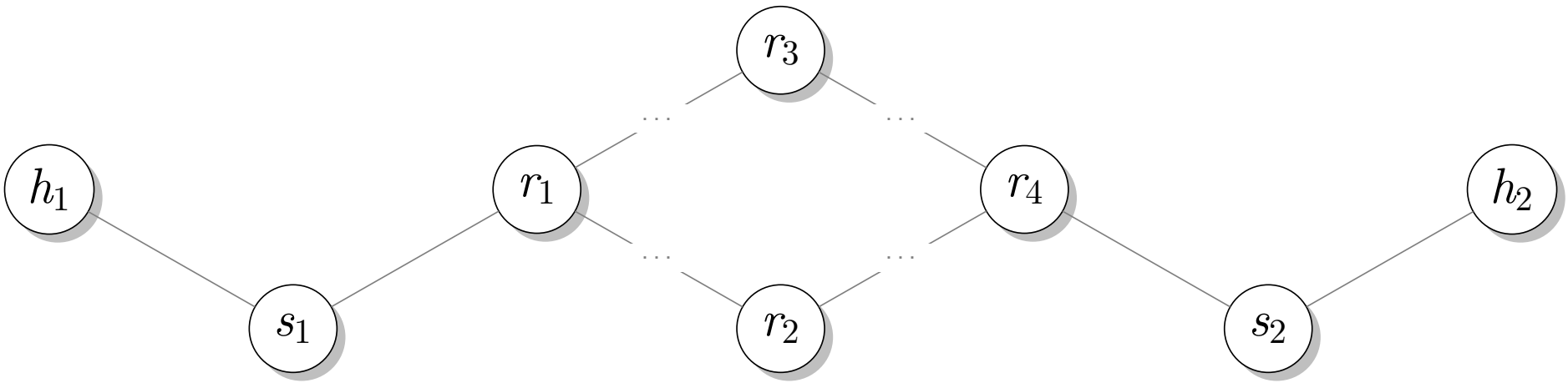
Hierarchic relations between flows couldn't be obtained from any data source, but might be inferred from the model built on basis of these data.

- The model should describe atomic communication units and relations between them.
- These atomic units shouldn't be tied to any specific protocol stack.
- A communication unit is not only a data transfer process, but may be a business process.
- The model should allow to map these units to network graph nodes.
- Using this model we could express data flows by given direction.
- For any given flow we could find all of its sub-flows.
- For any two given flows we could say if one is aggregated to another.

Network Nodes

- We are considering a network area N.

$$N = \{h_1, s_1, r_1, r_2, r_3, r_4, s_2, h_2\}$$



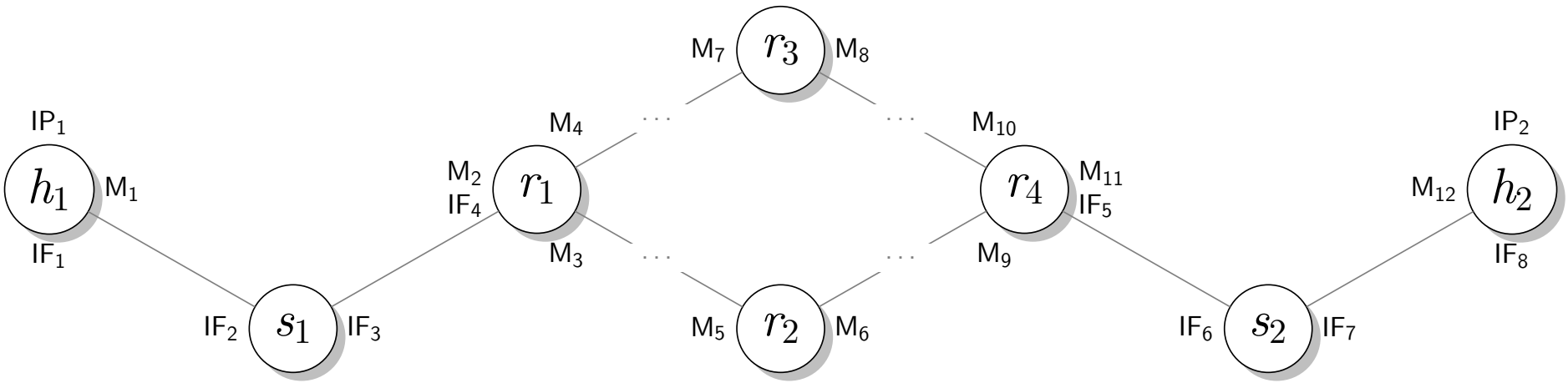
Network Nodes

- We are considering a network area N.

$$N = \{h_1, s_1, r_1, r_2, r_3, r_4, s_2, h_2\}$$

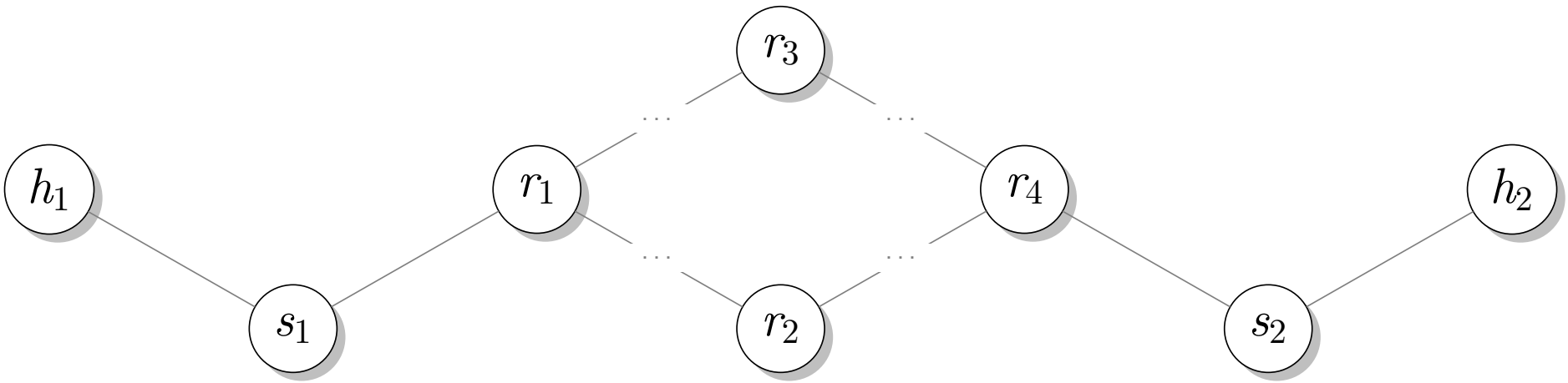
- During any time period τ each node $d \in N$ has a set of associated addresses $E^{\tau}(d)$.

$$E^{\tau}(h_1) = \{IP_1, M_1, IF_1\}, \quad E^{\tau}(s_1) = \{IF_2, IF_3\}, \quad \dots$$



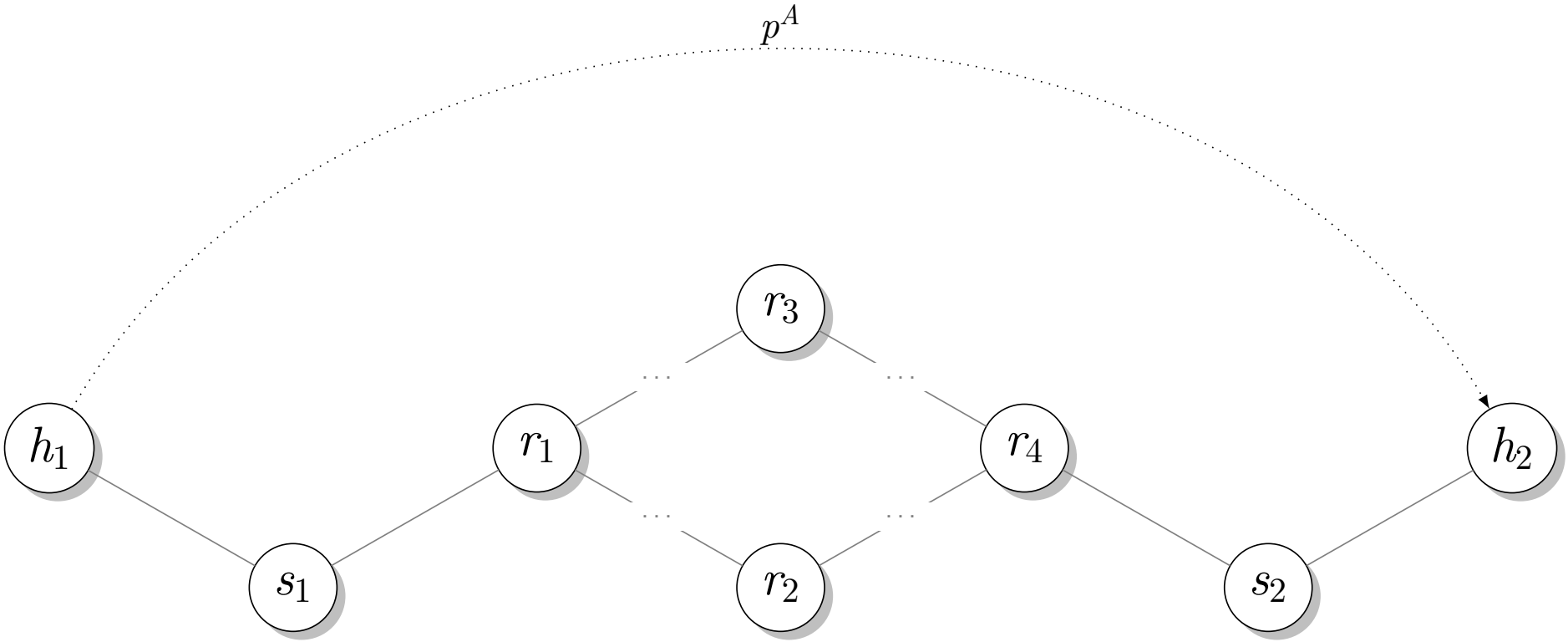
Telecommunication Processes (TPs)

- Communication between any nodes is a series of message exchanges.
- Each message is fragmented into chunks and passed using some protocol stack.
- Passing of each chunk is a data communication process.
- $P^{\tau}(N)$ is a set of all TPs executed during τ between nodes from set N.



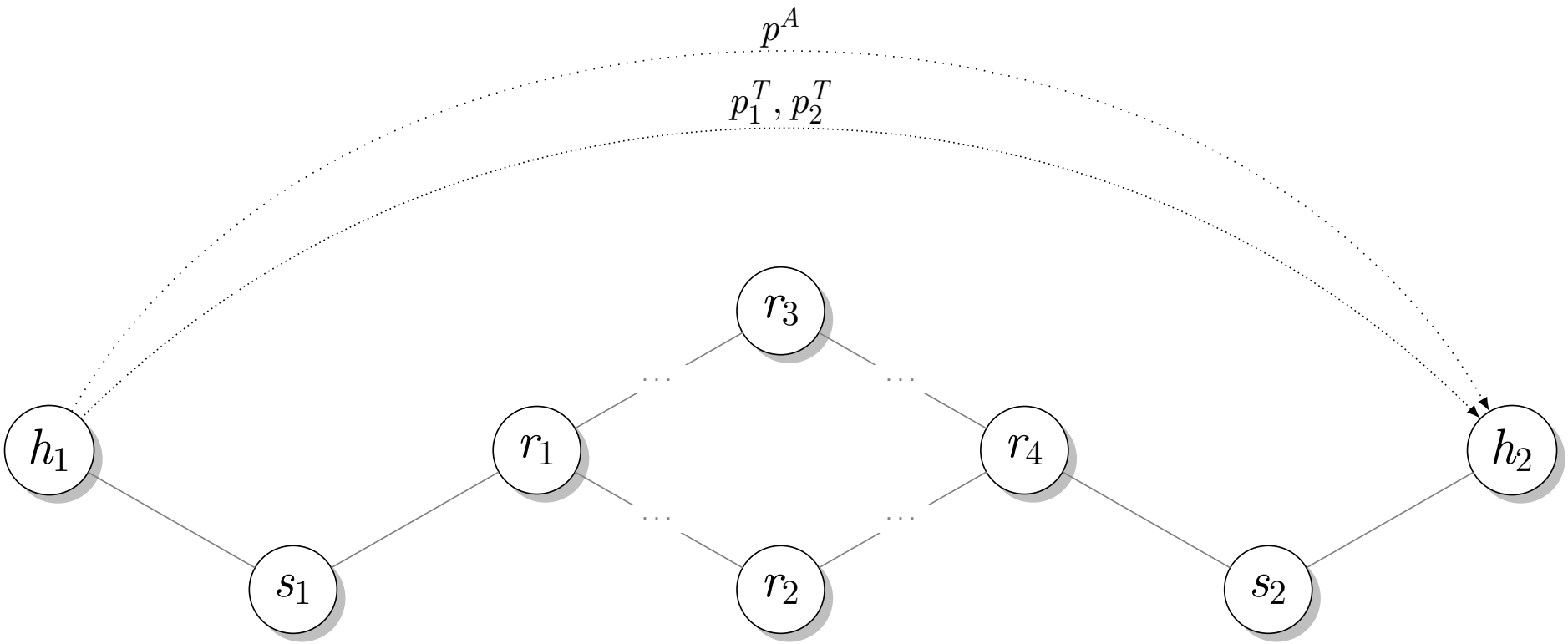
Telecommunication Processes (TPs)

- Communication between any nodes is a series of message exchanges.
- Each message is fragmented into chunks and passed using some protocol stack.
- Passing of each chunk is a data communication process.
- $P^{\tau}(N)$ is a set of all TPs executed during τ between nodes from set N.



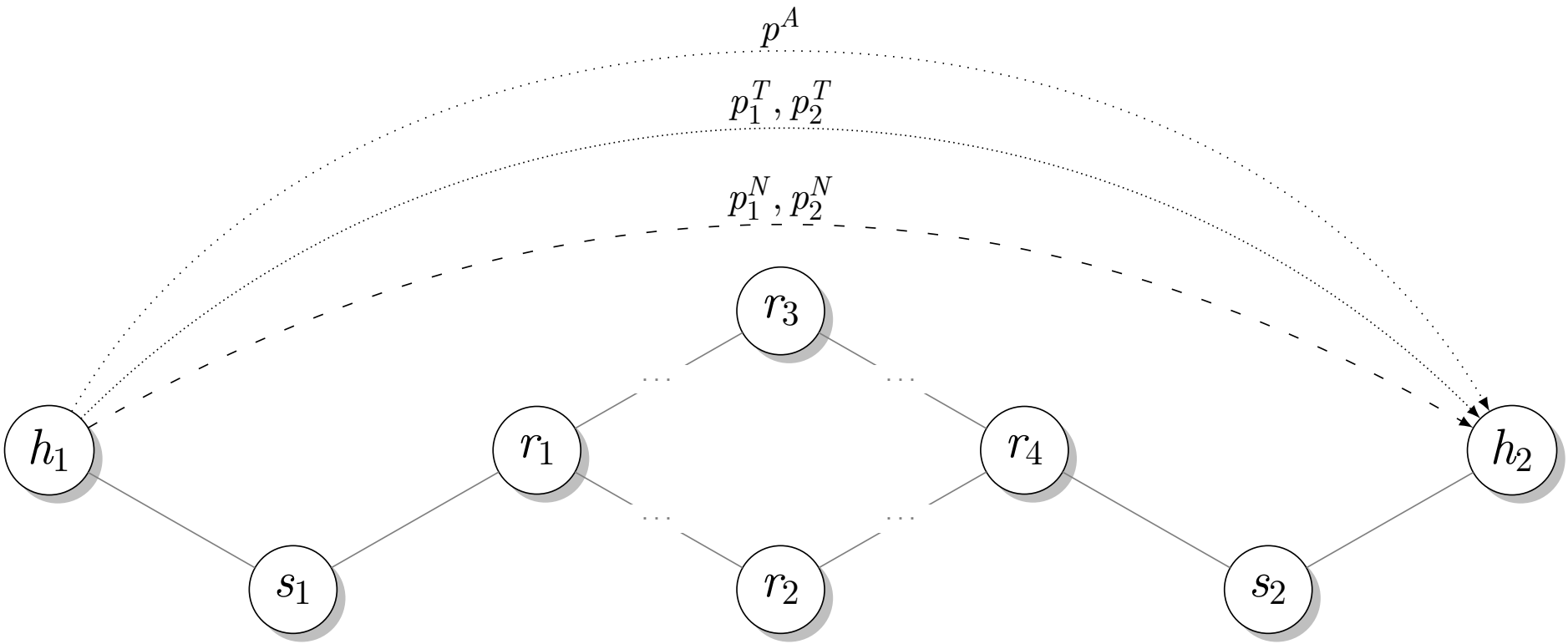
Telecommunication Processes (TPs)

- Communication between any nodes is a series of message exchanges.
- Each message is fragmented into chunks and passed using some protocol stack.
- Passing of each chunk is a data communication process.
- $P^{\tau}(N)$ is a set of all TPs executed during τ between nodes from set N.



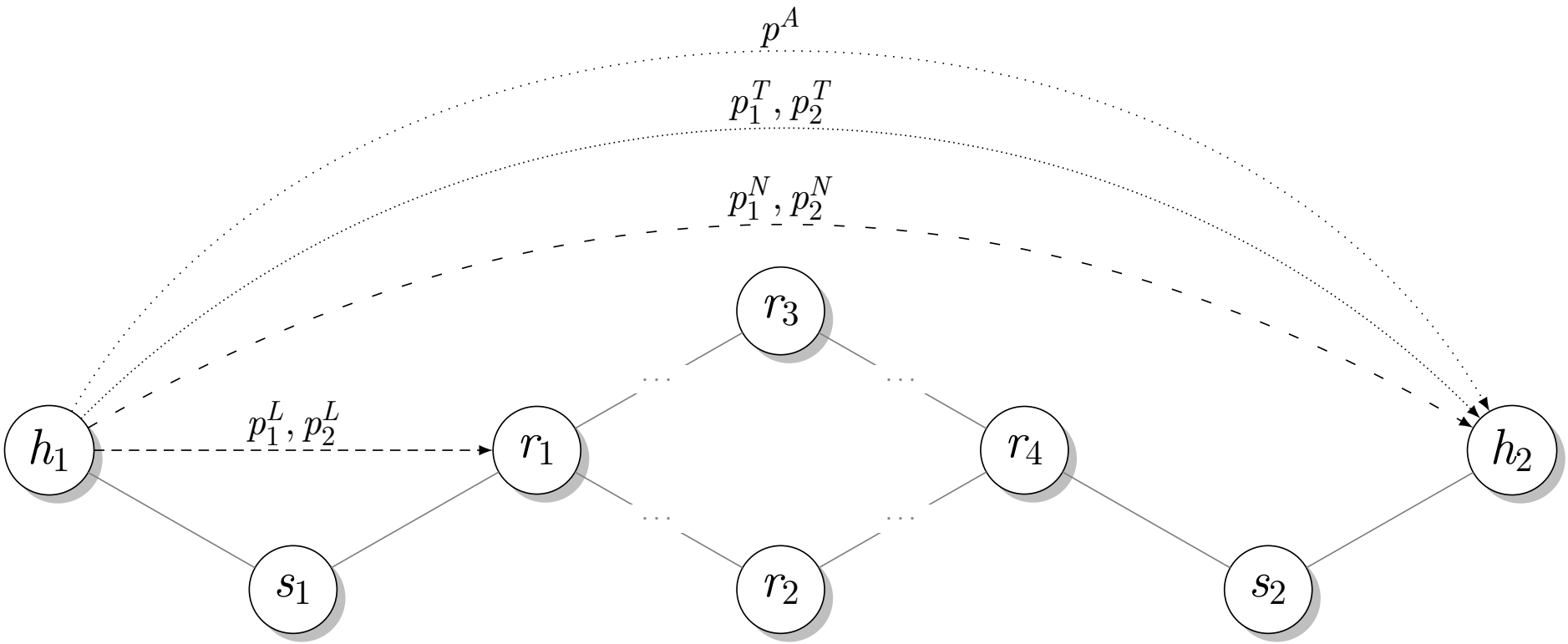
Telecommunication Processes (TPs)

- Communication between any nodes is a series of message exchanges.
- Each message is fragmented into chunks and passed using some protocol stack.
- Passing of each chunk is a data communication process.
- $P^{\tau}(N)$ is a set of all TPs executed during τ between nodes from set N.



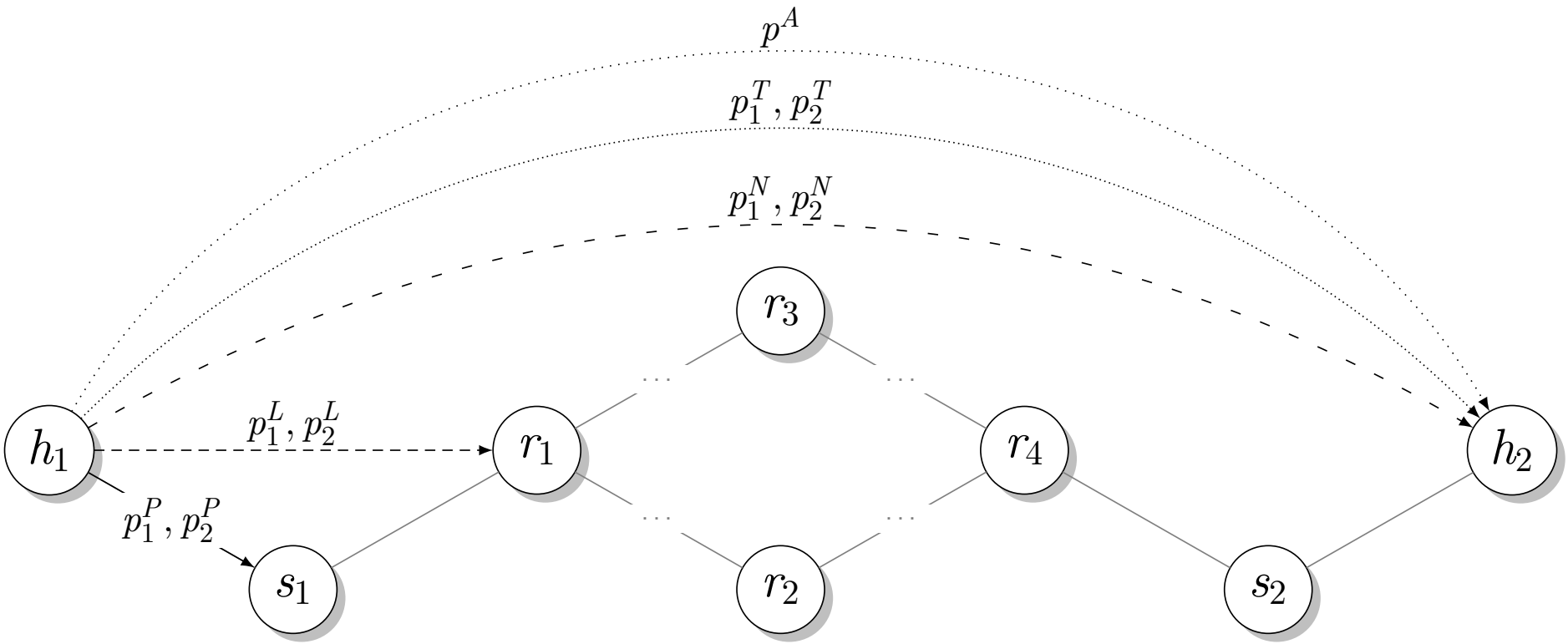
Telecommunication Processes (TPs)

- Communication between any nodes is a series of message exchanges.
- Each message is fragmented into chunks and passed using some protocol stack.
- Passing of each chunk is a data communication process.
- $P^{\tau}(N)$ is a set of all TPs executed during τ between nodes from set N.



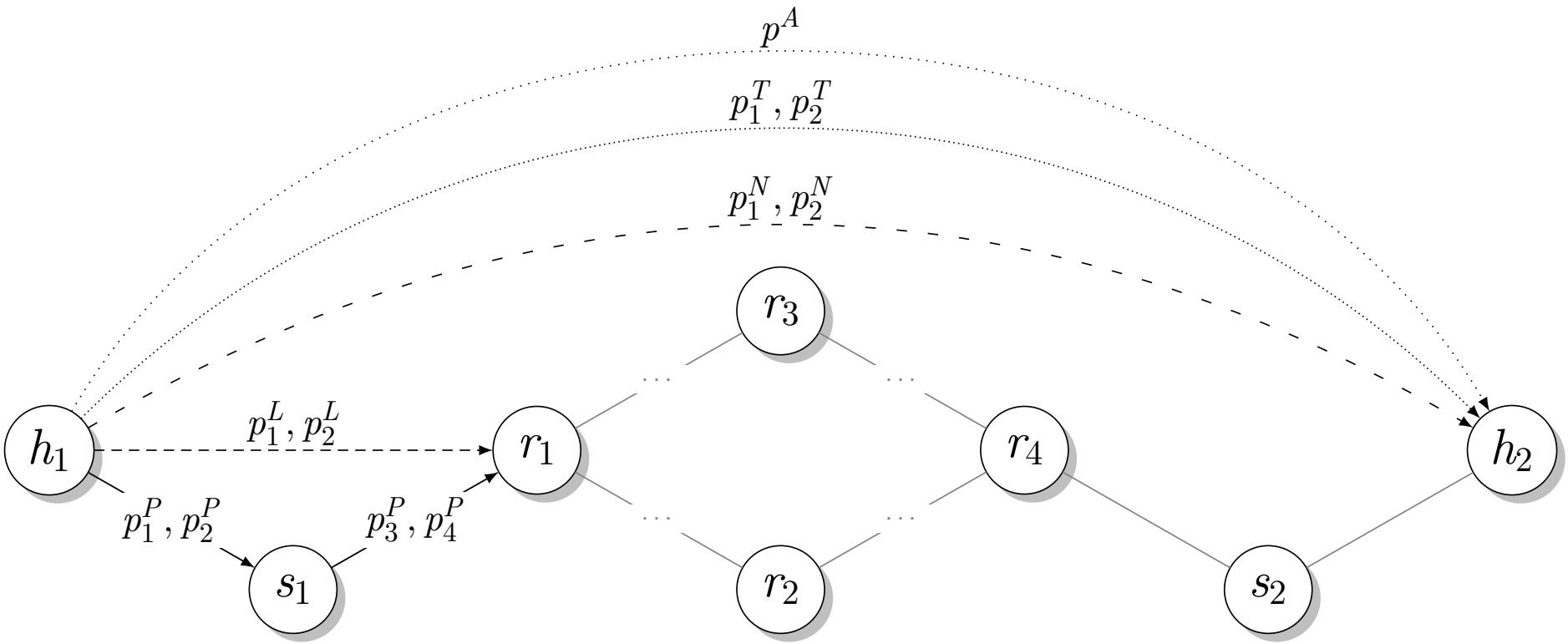
Telecommunication Processes (TPs)

- Communication between any nodes is a series of message exchanges.
- Each message is fragmented into chunks and passed using some protocol stack.
- Passing of each chunk is a data communication process.
- $P^{\tau}(N)$ is a set of all TPs executed during τ between nodes from set N.



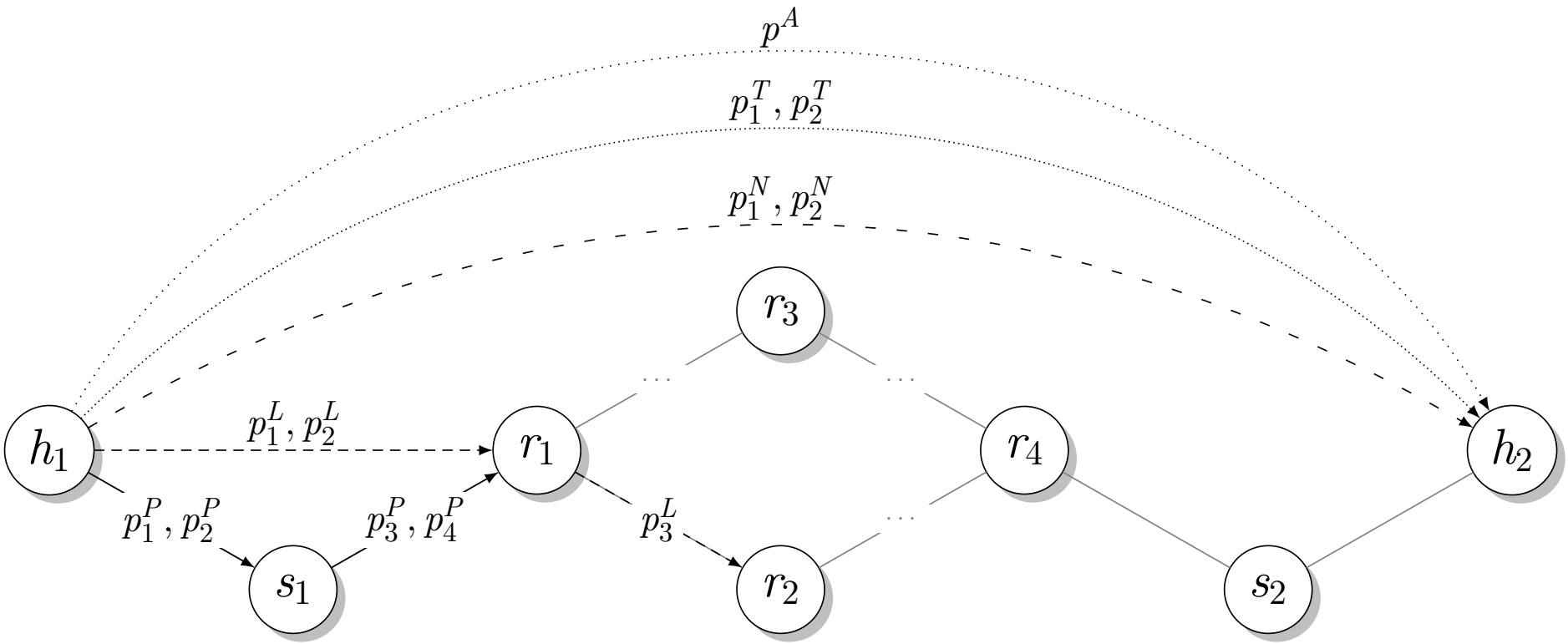
Telecommunication Processes (TPs)

- Communication between any nodes is a series of message exchanges.
- Each message is fragmented into chunks and passed using some protocol stack.
- Passing of each chunk is a data communication process.
- $P^{\tau}(N)$ is a set of all TPs executed during τ between nodes from set N.



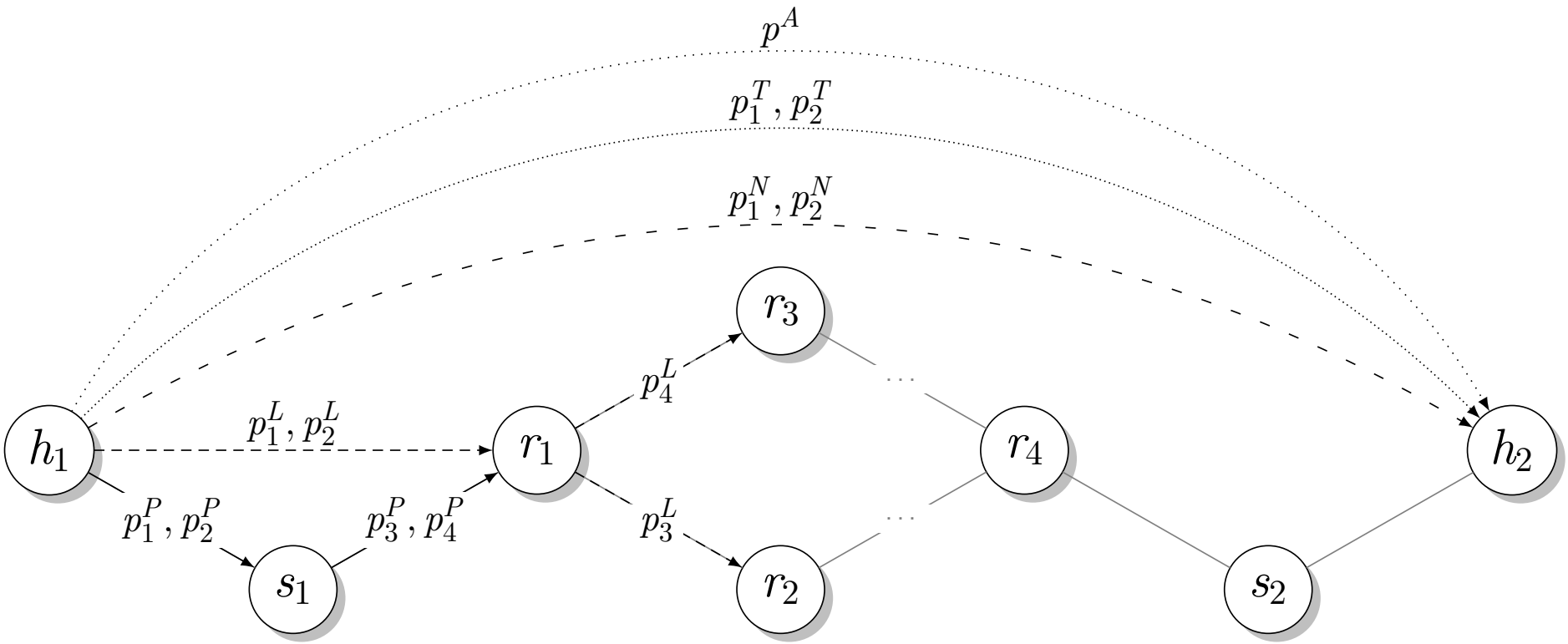
Telecommunication Processes (TPs)

- Communication between any nodes is a series of message exchanges.
- Each message is fragmented into chunks and passed using some protocol stack.
- Passing of each chunk is a data communication process.
- $P^{\tau}(N)$ is a set of all TPs executed during τ between nodes from set N.



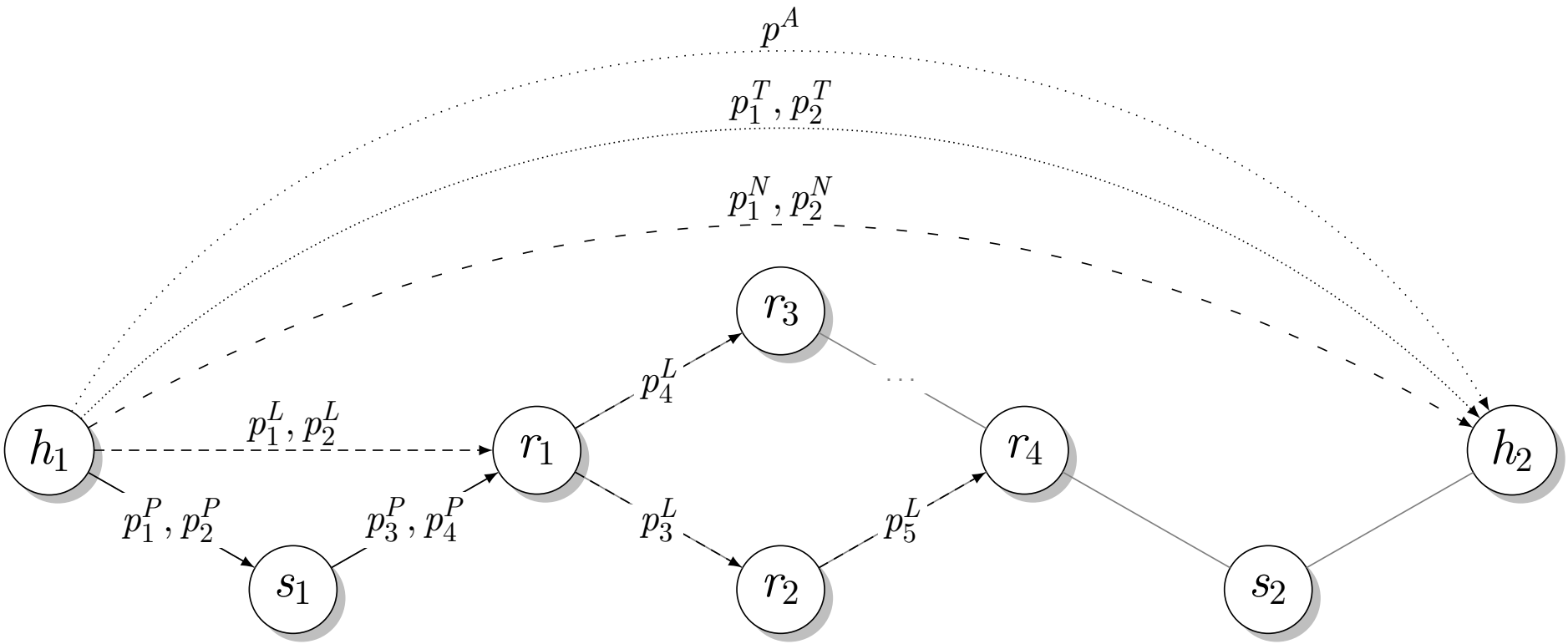
Telecommunication Processes (TPs)

- Communication between any nodes is a series of message exchanges.
- Each message is fragmented into chunks and passed using some protocol stack.
- Passing of each chunk is a data communication process.
- $P^{\tau}(N)$ is a set of all TPs executed during τ between nodes from set N.



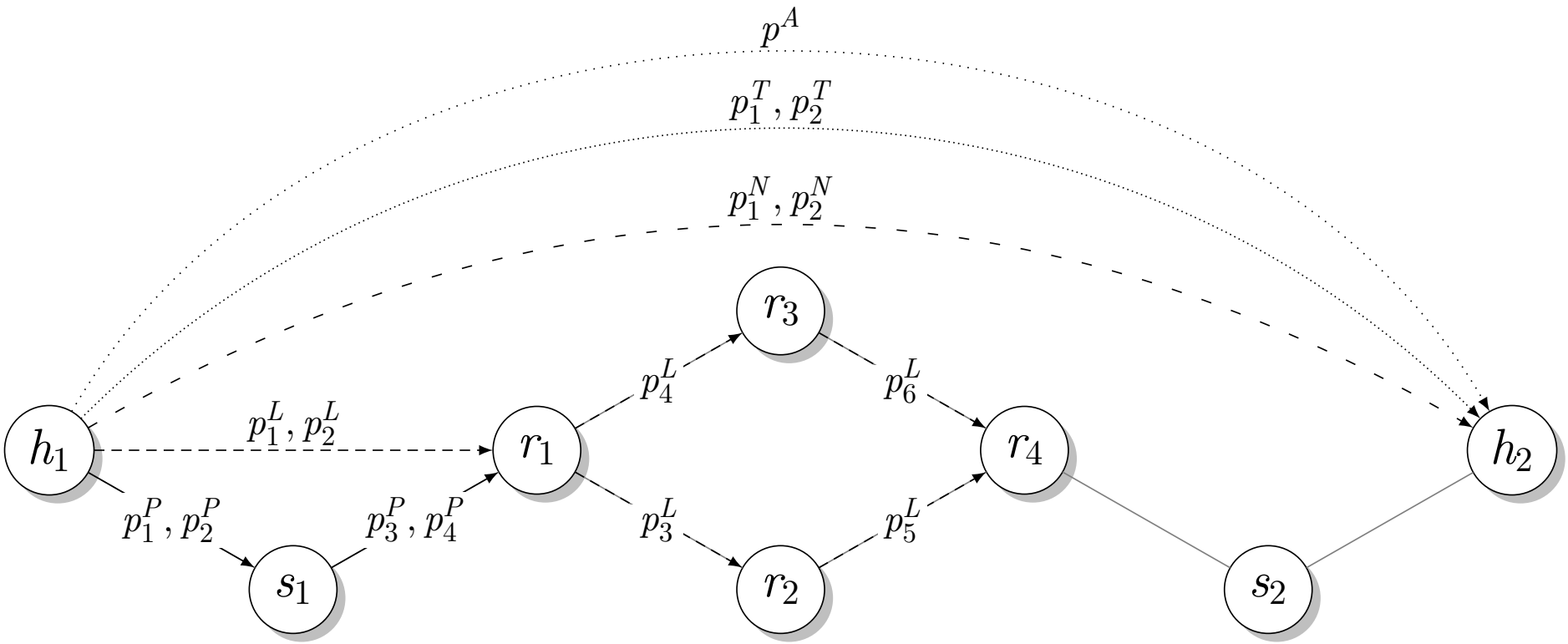
Telecommunication Processes (TPs)

- Communication between any nodes is a series of message exchanges.
- Each message is fragmented into chunks and passed using some protocol stack.
- Passing of each chunk is a data communication process.
- $P^{\tau}(N)$ is a set of all TPs executed during τ between nodes from set N.



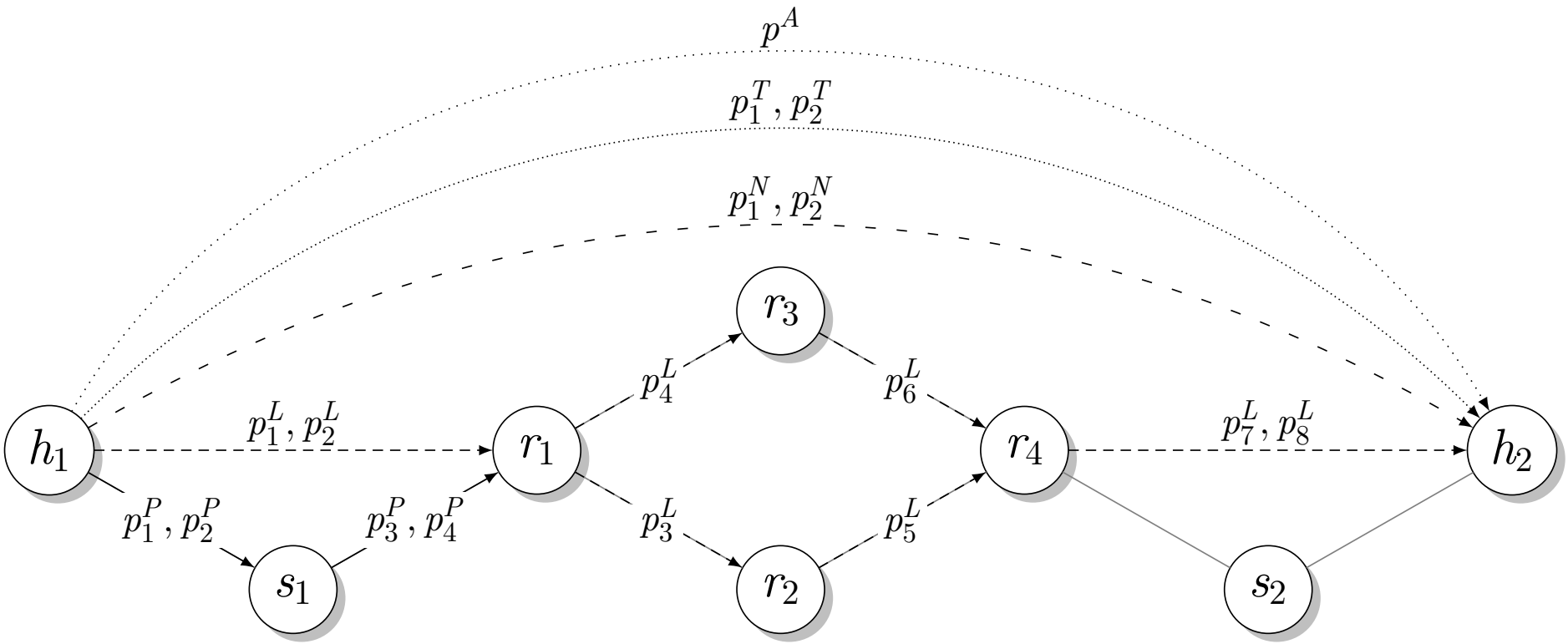
Telecommunication Processes (TPs)

- Communication between any nodes is a series of message exchanges.
- Each message is fragmented into chunks and passed using some protocol stack.
- Passing of each chunk is a data communication process.
- $P^{\tau}(N)$ is a set of all TPs executed during τ between nodes from set N.



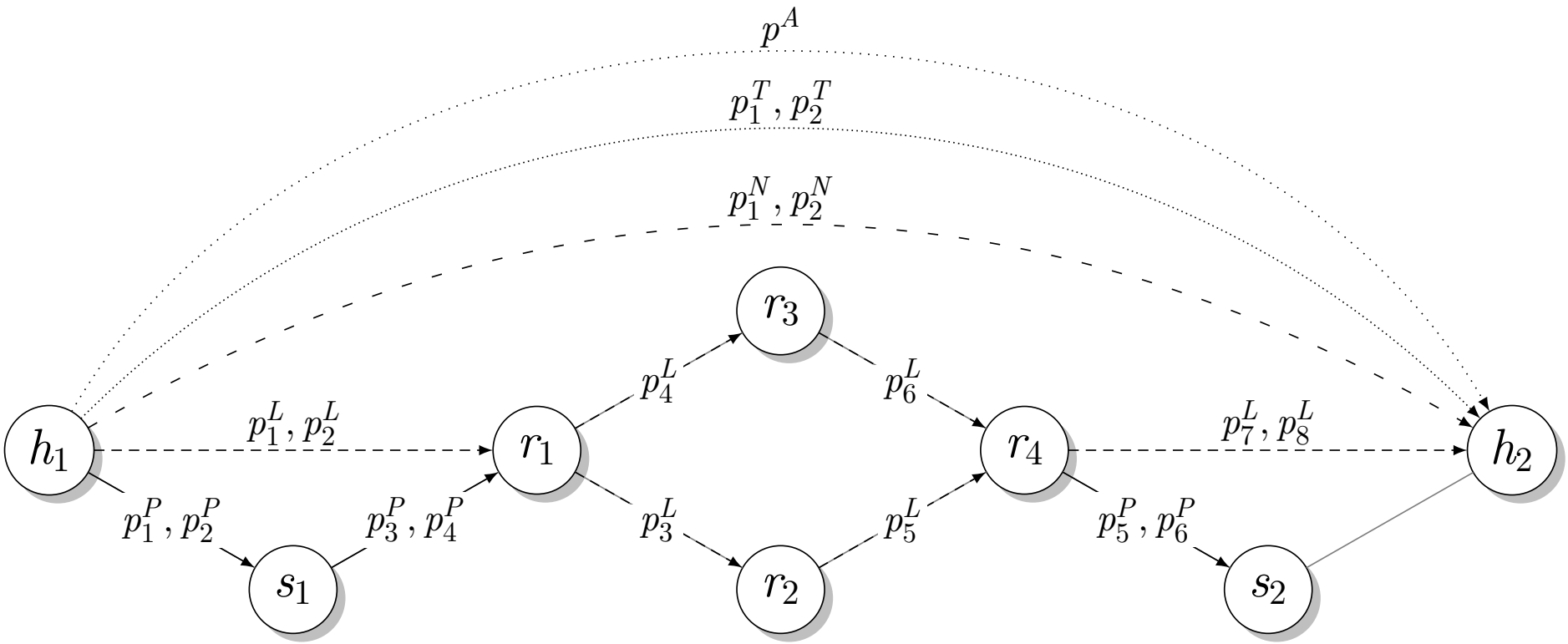
Telecommunication Processes (TPs)

- Communication between any nodes is a series of message exchanges.
- Each message is fragmented into chunks and passed using some protocol stack.
- Passing of each chunk is a data communication process.
- $P^{\tau}(N)$ is a set of all TPs executed during τ between nodes from set N.



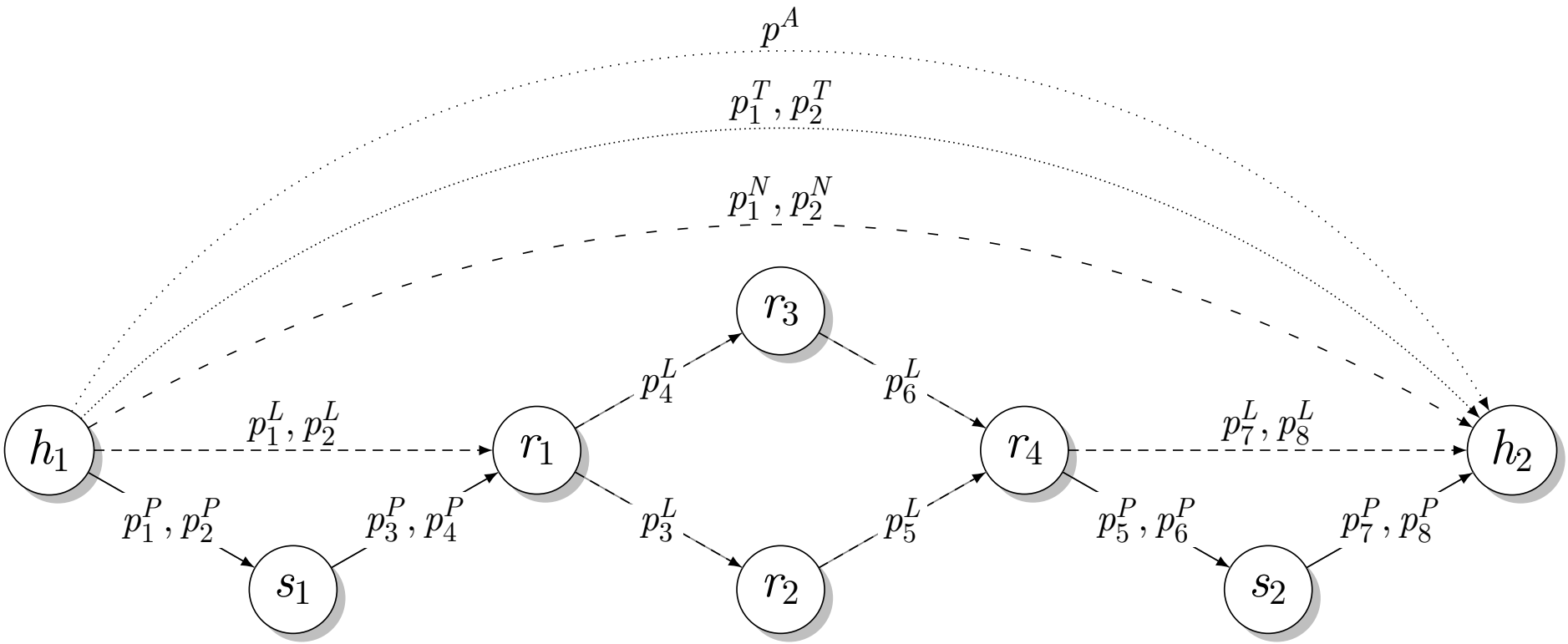
Telecommunication Processes (TPs)

- Communication between any nodes is a series of message exchanges.
- Each message is fragmented into chunks and passed using some protocol stack.
- Passing of each chunk is a data communication process.
- $P^{\tau}(N)$ is a set of all TPs executed during τ between nodes from set N.



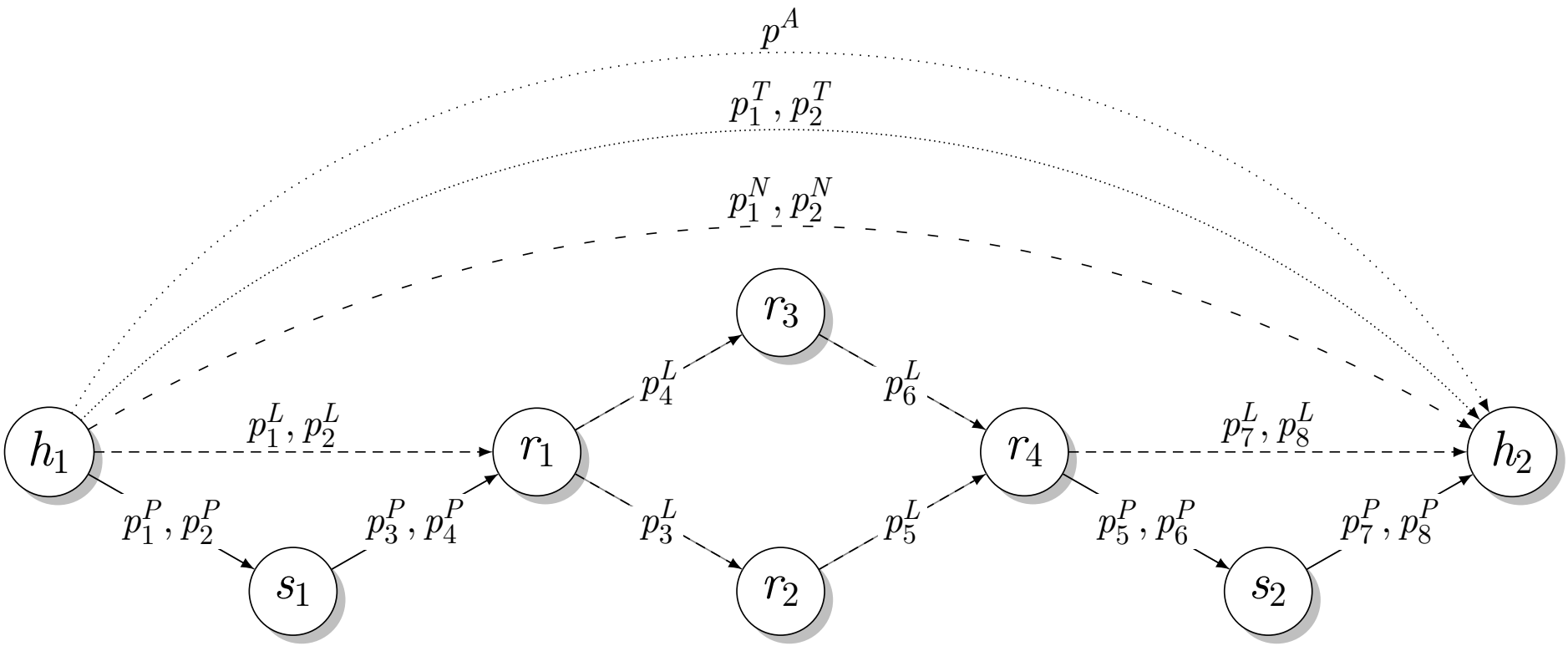
Telecommunication Processes (TPs)

- Communication between any nodes is a series of message exchanges.
- Each message is fragmented into chunks and passed using some protocol stack.
- Passing of each chunk is a data communication process.
- $P^{\tau}(N)$ is a set of all TPs executed during τ between nodes from set N.



Telecommunication Process Properties

- Each process $p \in P^{\tau}(N)$ is characterized by:
  - source address $s(p) \in E^{\tau}(N)$;
  - destination addresses $d(p) \in E^{\tau}(N)$;
  - a set of attributes $a(p)$;
  - timestamps of the process start $t_s(p) \in \tau$ and the process end $t_e(p) \in \tau$.

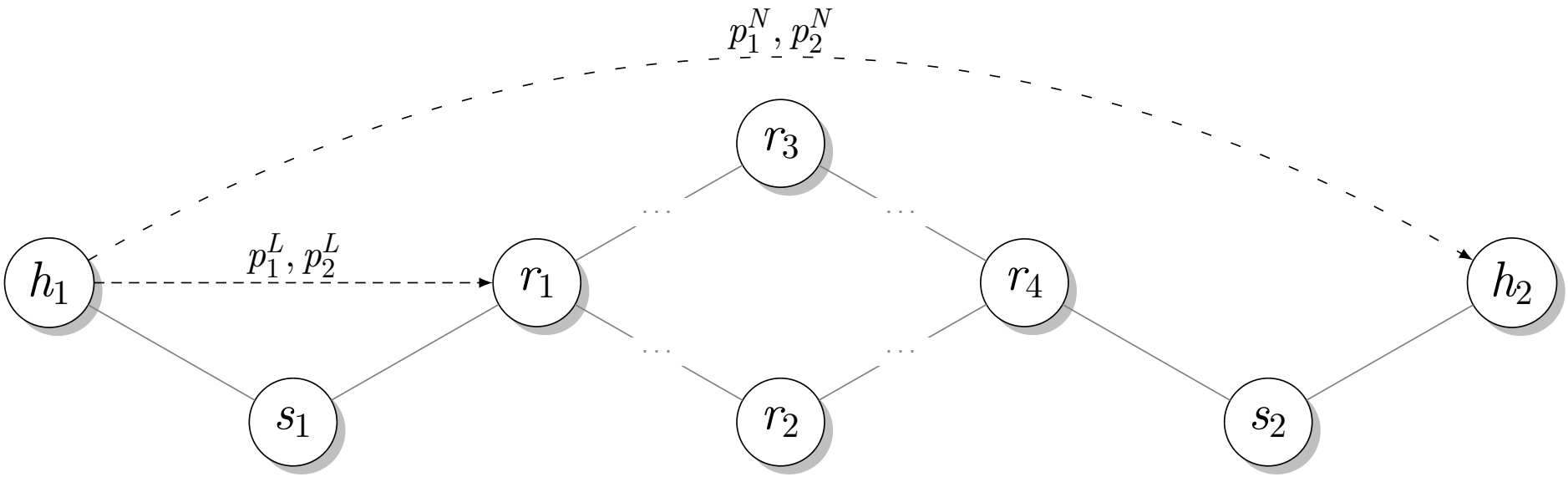


TPs Hierarchy Tree

- Each process p during its execution can generate another process:

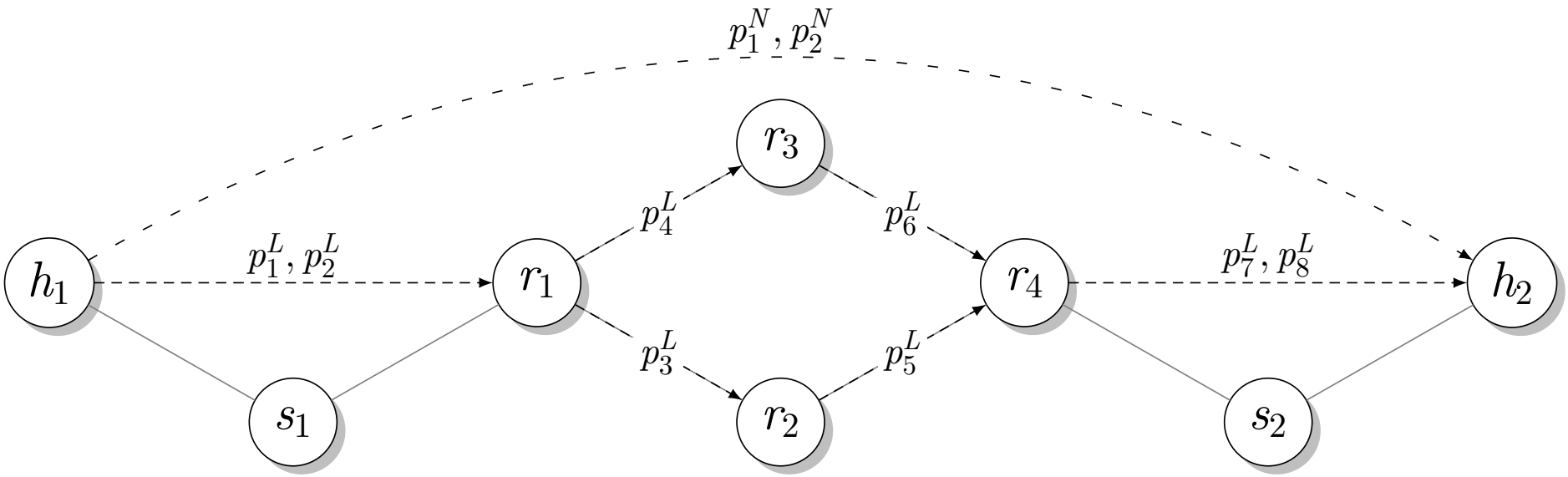
TPs Hierarchy Tree

- Each process p during its execution can generate another process:
  1. if it can not transfer a data block to the endpoint directly, then a *child* process q is generated (p is called the *parent* of q);



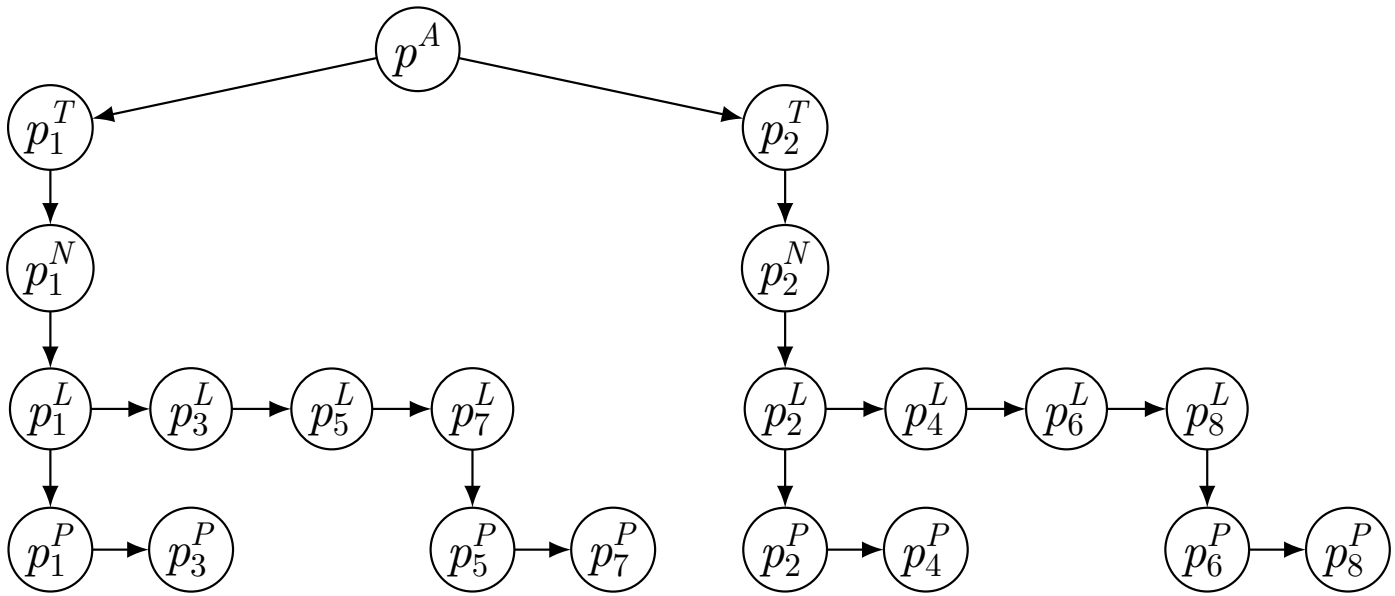
TPs Hierarchy Tree

- Each process p during its execution can generate another process:
  1. if it can not transfer a data block to the endpoint directly, then a *child* process q is generated (p is called the *parent* of q);
  2. if the endpoint of the process doesn't coincide with the endpoint of the parent process, then a *subsequent* process q is generated (p is called a *predecessor* of q).



TPs Hierarchy Tree

- Each process p during its execution can generate another process:
  1. if it can not transfer a data block to the endpoint directly, then a *child* process q is generated (p is called the *parent* of q);
  2. if the endpoint of the process doesn't coincide with the endpoint of the parent process, then a *subsequent* process q is generated (p is called a *predecessor* of q).
- Using these rules we can build a tree of TPs hierarchy.



Traffic Flows

Flow definition

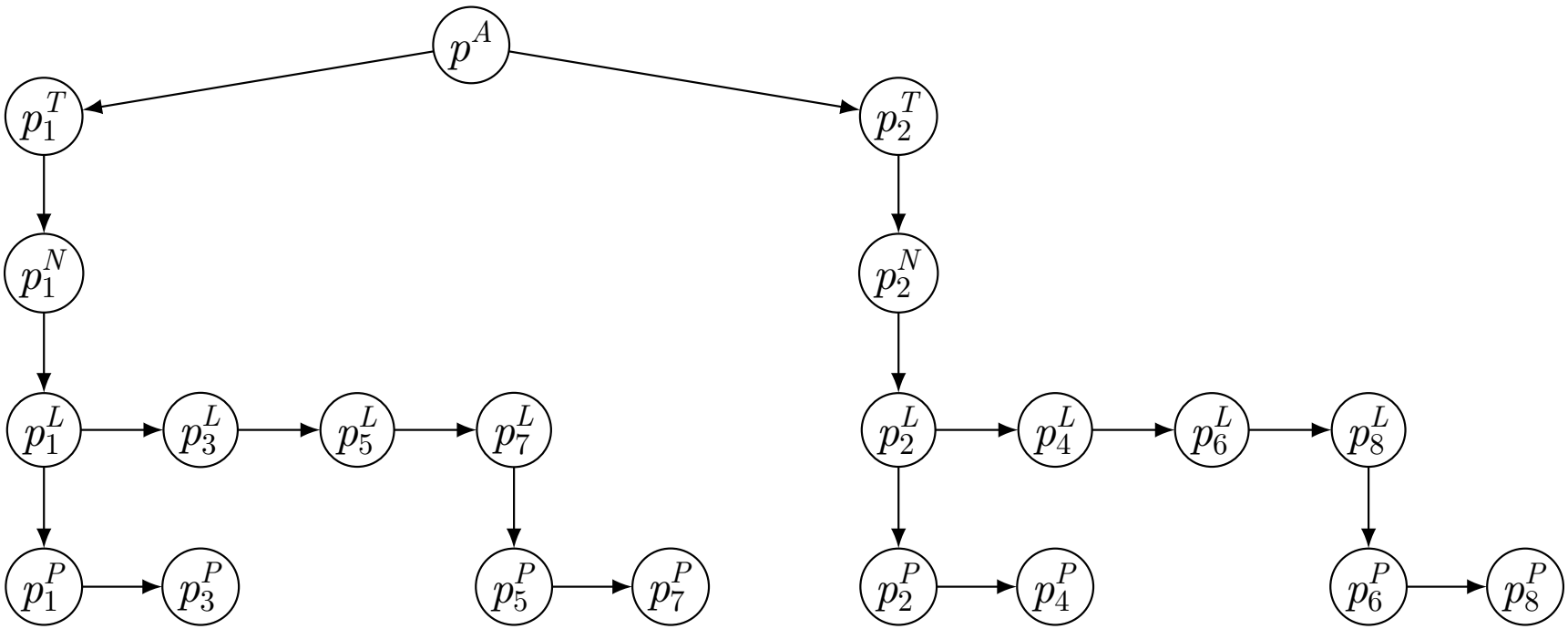
A traffic flow is a set of data blocks (packets, frames, messages, ...) passing a network during a certain time interval and having a set of common properties.

- Flow doesn't exist in the network, it is just a slice of traffic, defined by a network engineer.
- As each TP corresponds to a data block, so any subset of $P^{\tau}(N)$ corresponds to some flow during τ.
- To define a slice of traffic, engineer specifies a direction — a tuple, describing sources, destinations and attributes of the processes, carrying interesting traffic, e.g.:

$$\delta = \langle \{IP_1, M_3\}, \{IP_2\}, \{\langle vlan, 2 \rangle\} \rangle$$

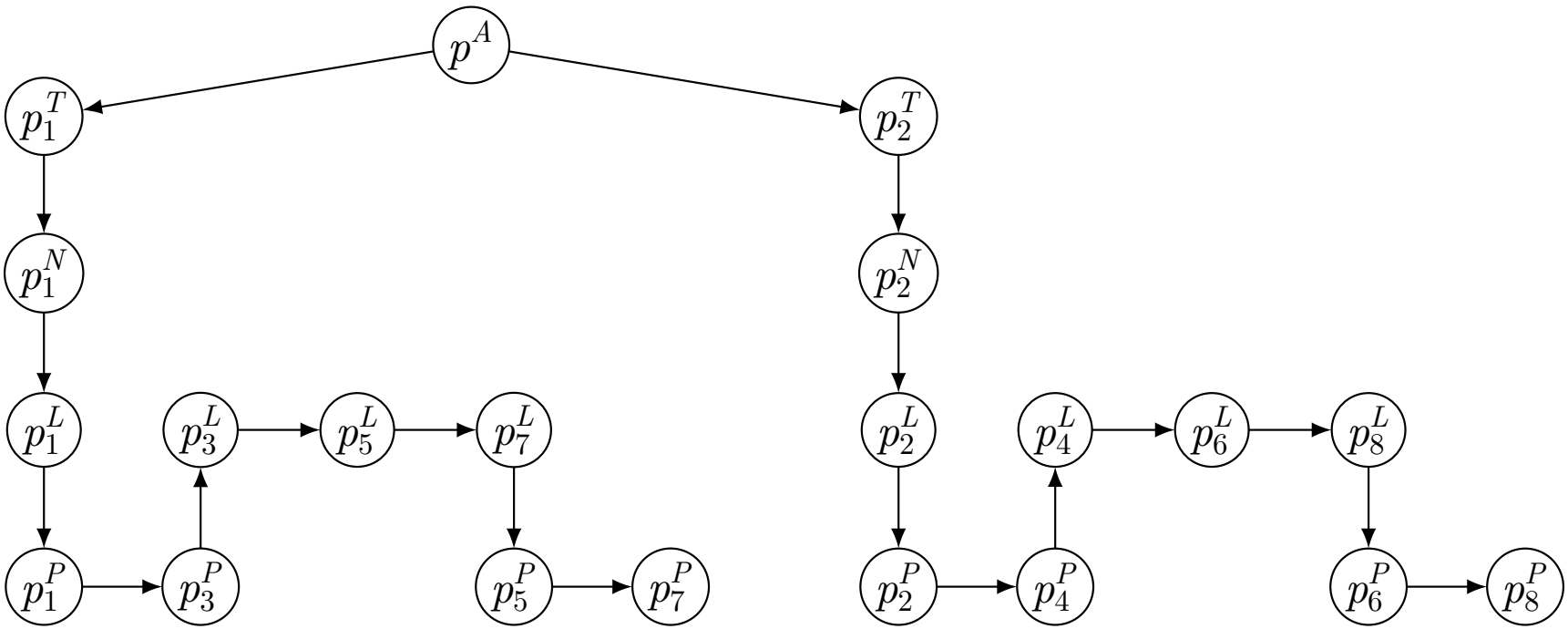
Traffic Flows (continuation)

- Flow by the given direction could be inferred from the TP hierarchy tree.



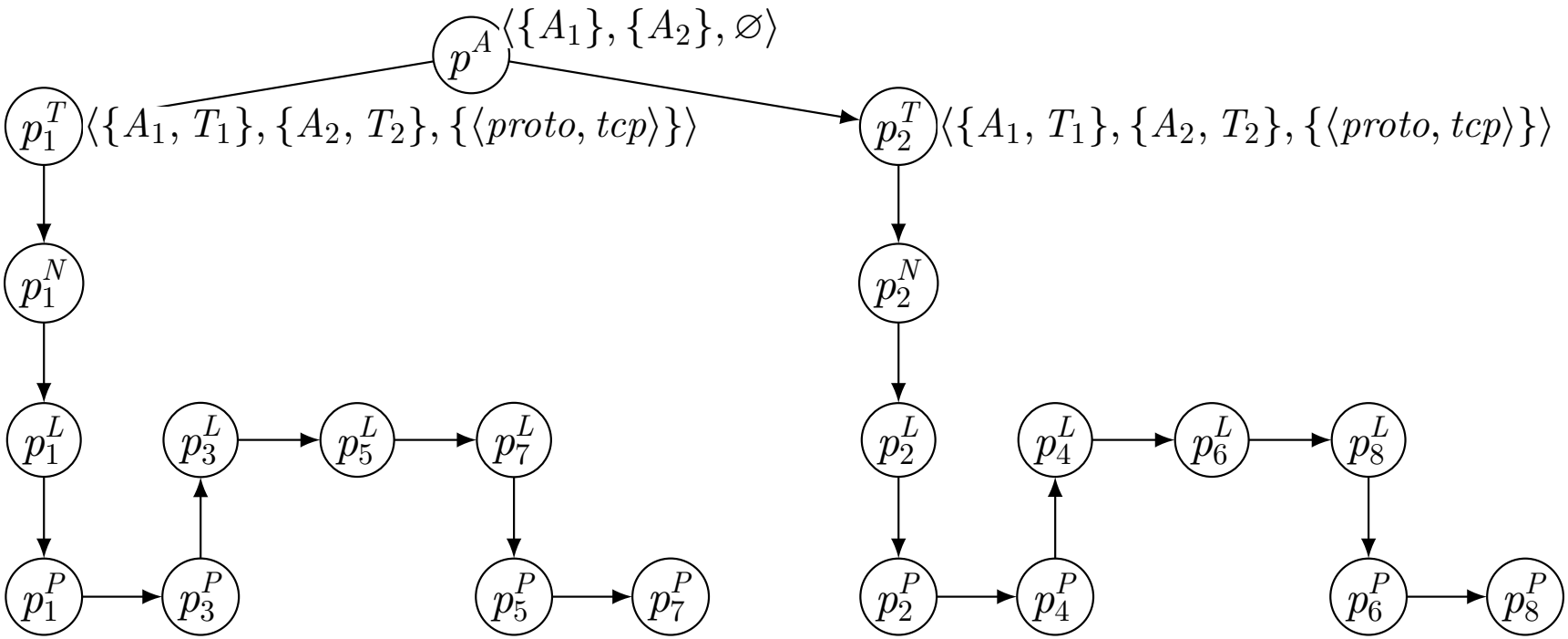
Traffic Flows (continuation)

- Flow by the given direction could be inferred from the TP hierarchy tree.
  - at first it should be slightly simplified.



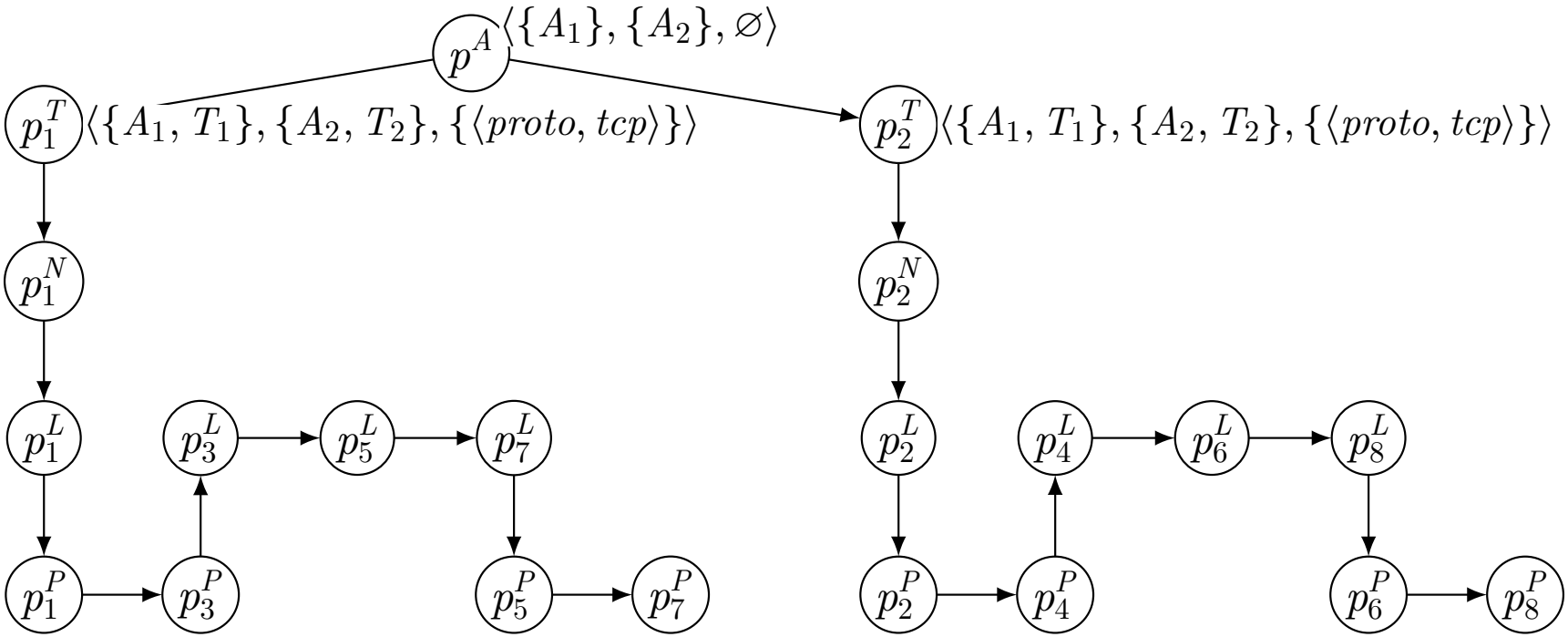
Traffic Flows (continuation)

- Flow by the given direction could be inferred from the TP hierarchy tree.
  - at first it should be slightly simplified.
- Each TP has a direction history: predecessor history + its own direction.

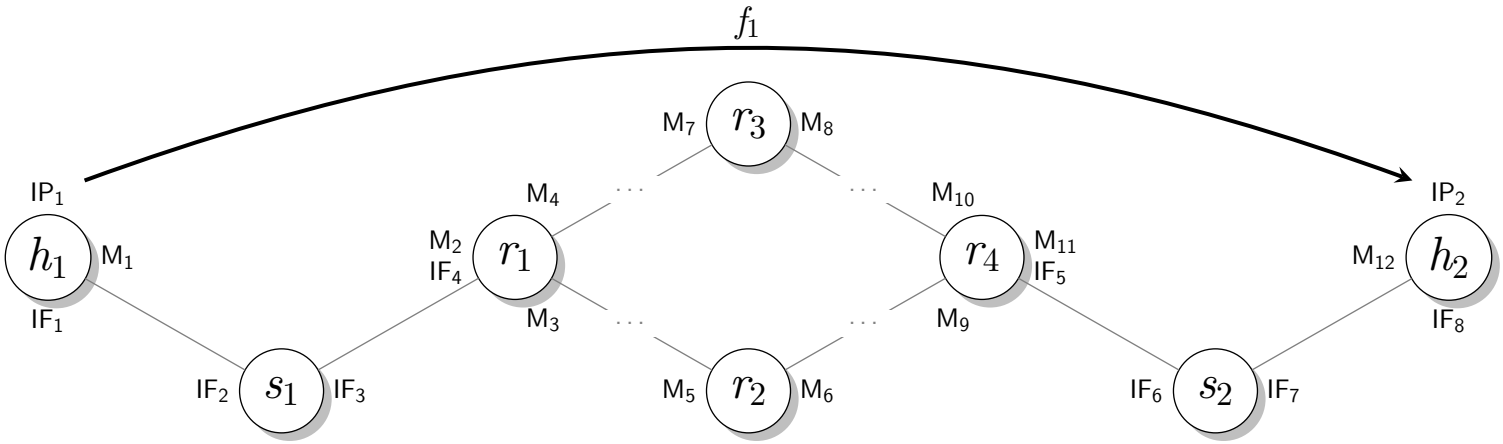


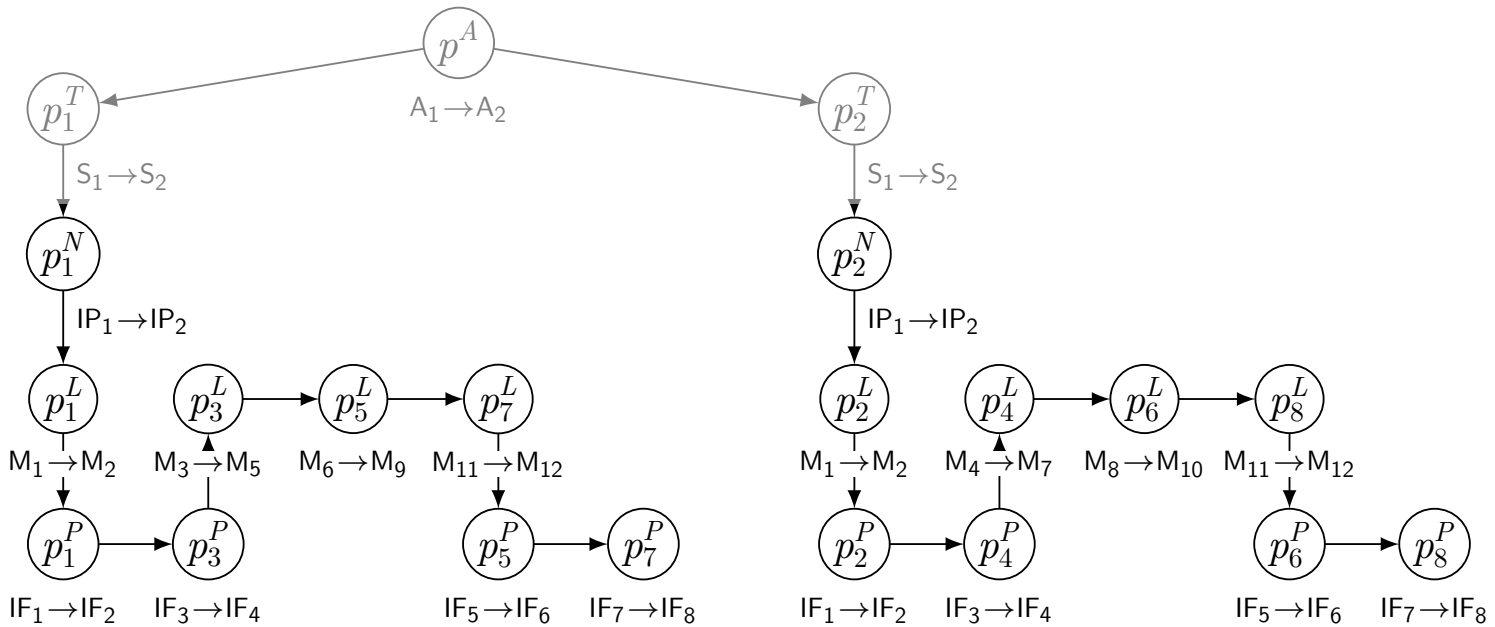
Traffic Flows (continuation)

- Flow by the given direction could be inferred from the TP hierarchy tree.
  - at first it should be slightly simplified.
- Each TP has a direction history: predecessor history + its own direction.
- Flow by some direction δ during time interval τ is a set of processes, which direction history include δ.

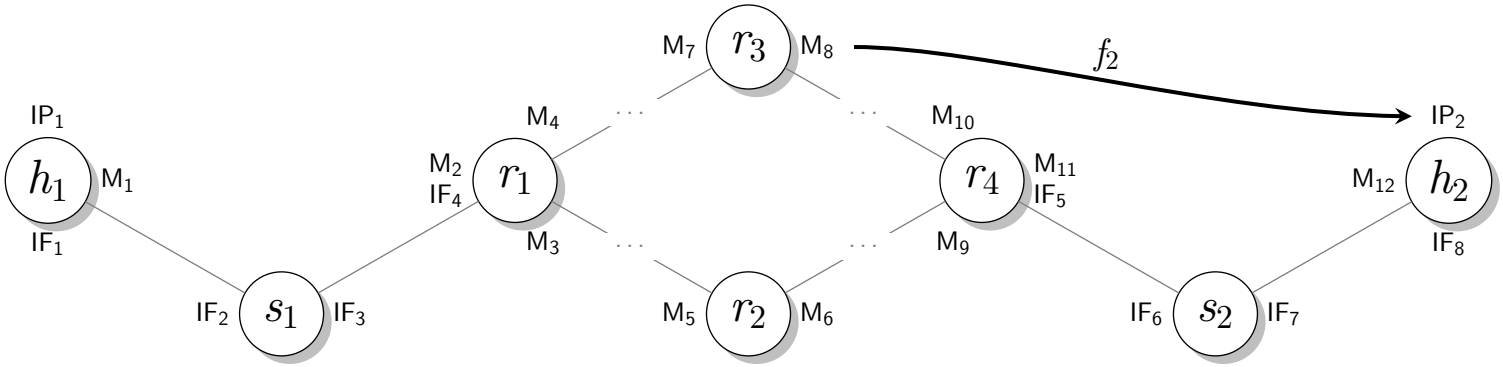


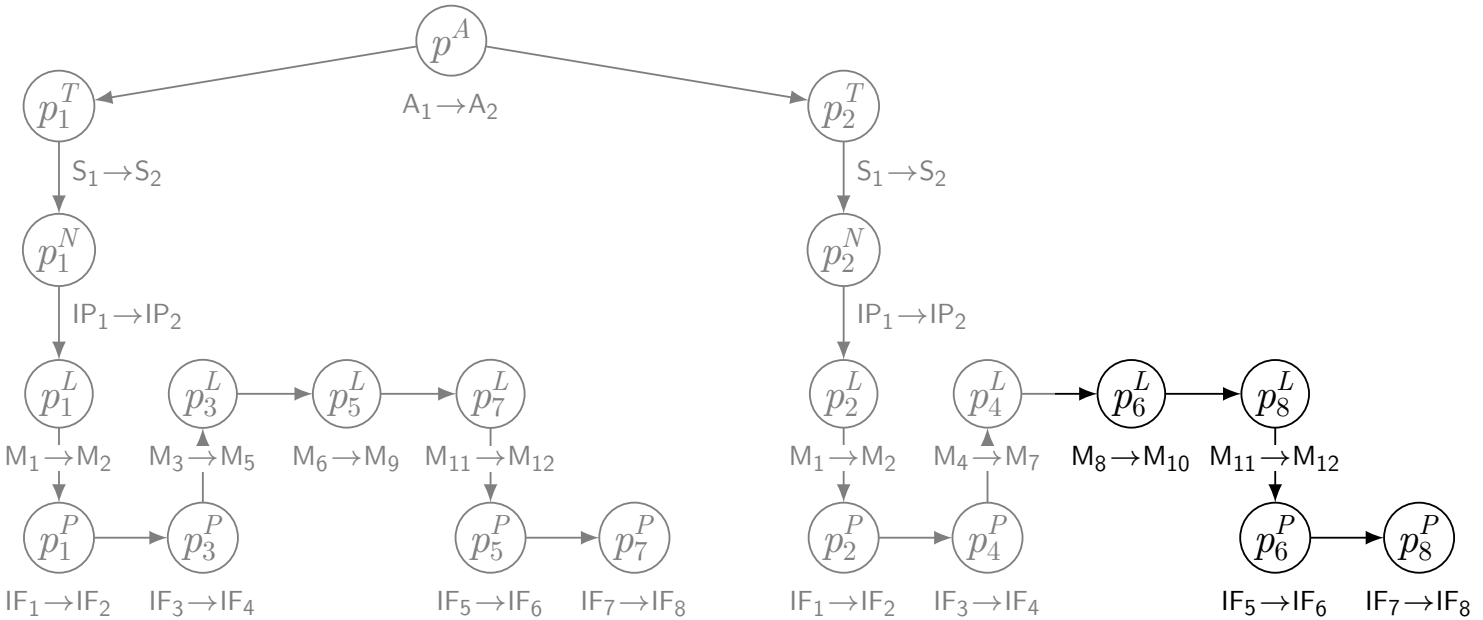
Flow Example 1: $f^{\tau}_{\langle\{IP_1\},\{IP_2\},\varnothing\rangle} = \{p_1^N, p_2^N\}$



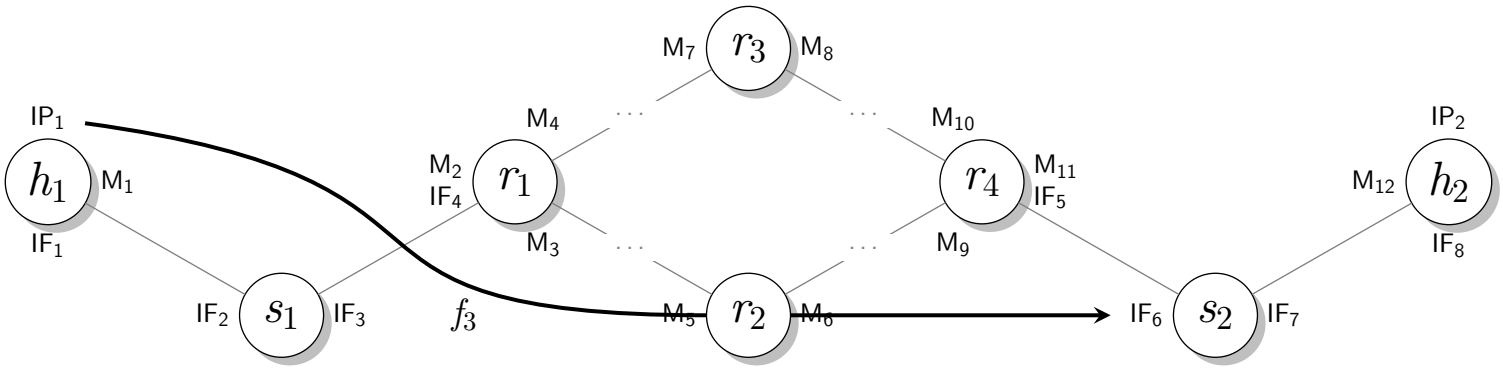


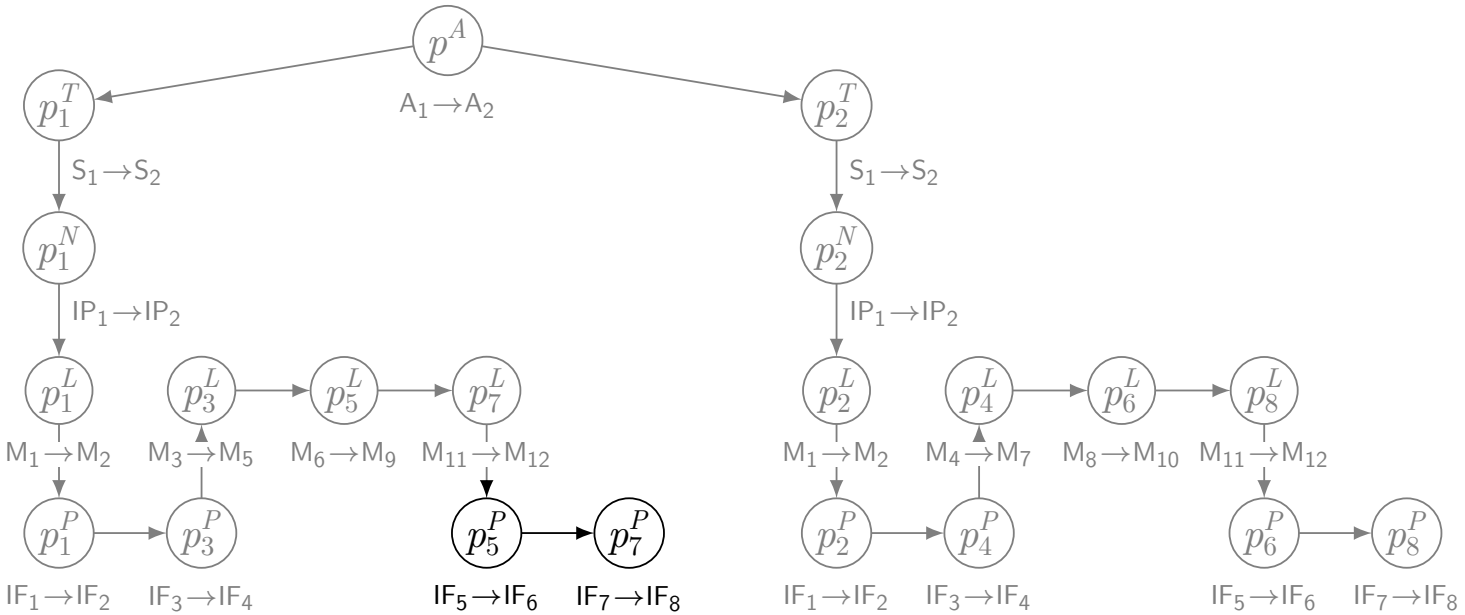
Flow Example 2: $f^{\tau}_{\langle\{M_8\},\{IP_2\},\varnothing\rangle} = \{p_6^L\}$



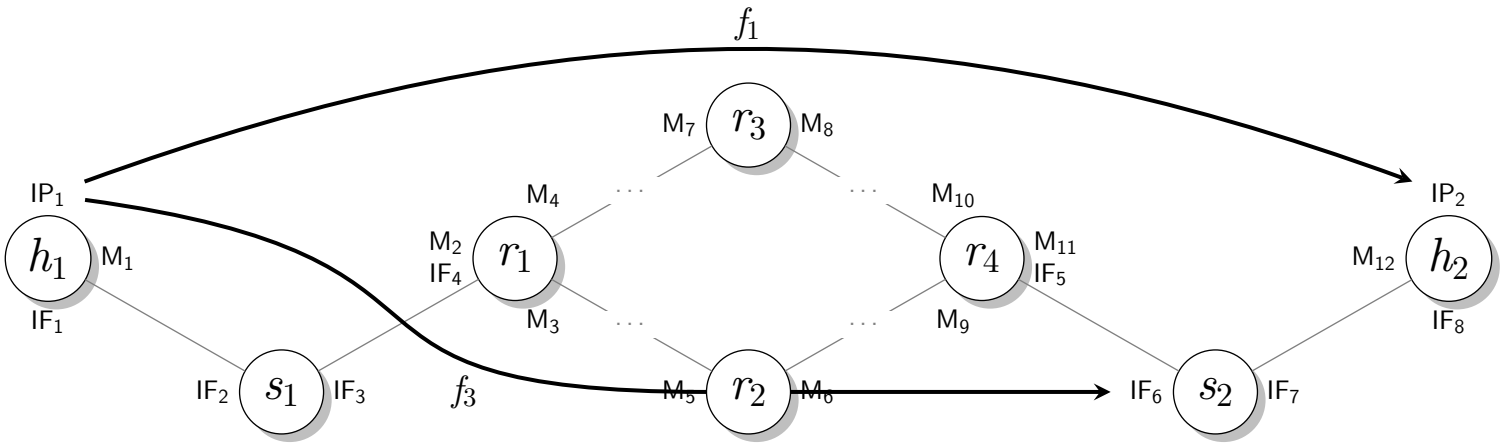


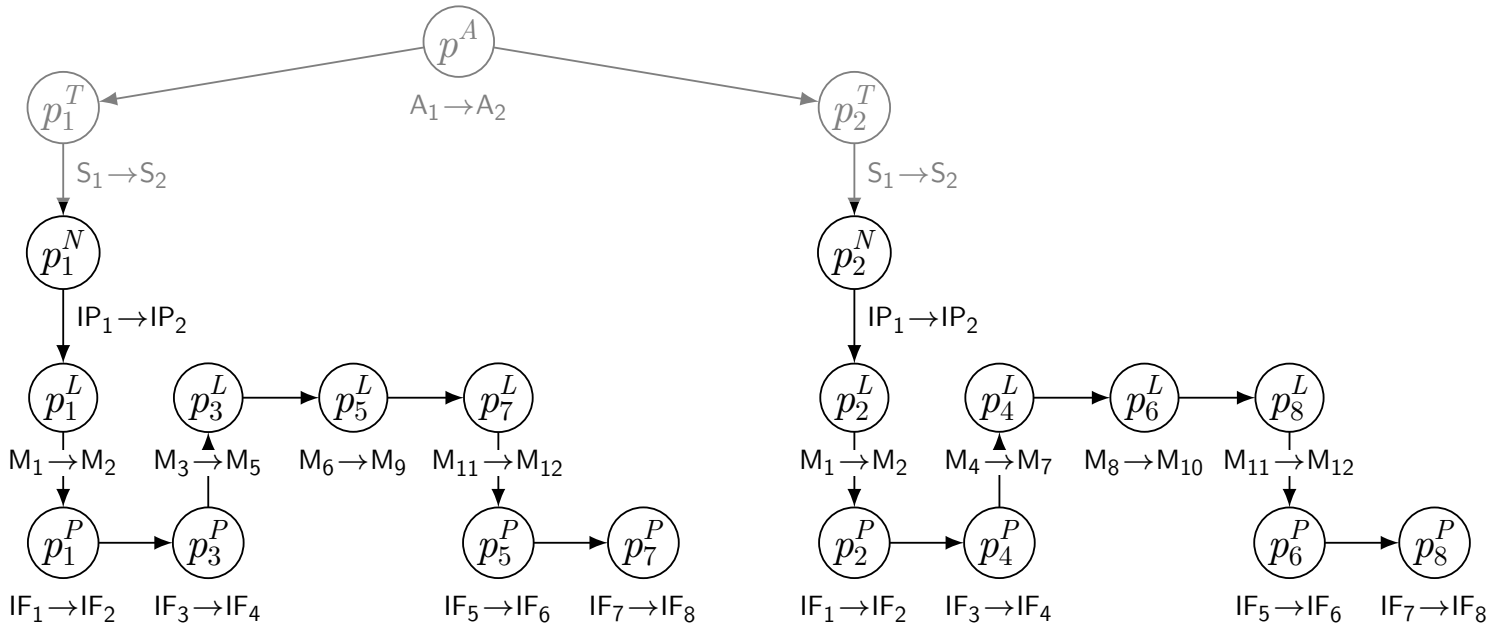
Flow Example 3: $f^{\tau}_{\langle\{IP_1\},\{M_5,IF_6\},\varnothing\rangle} = \{p_5^P\}$



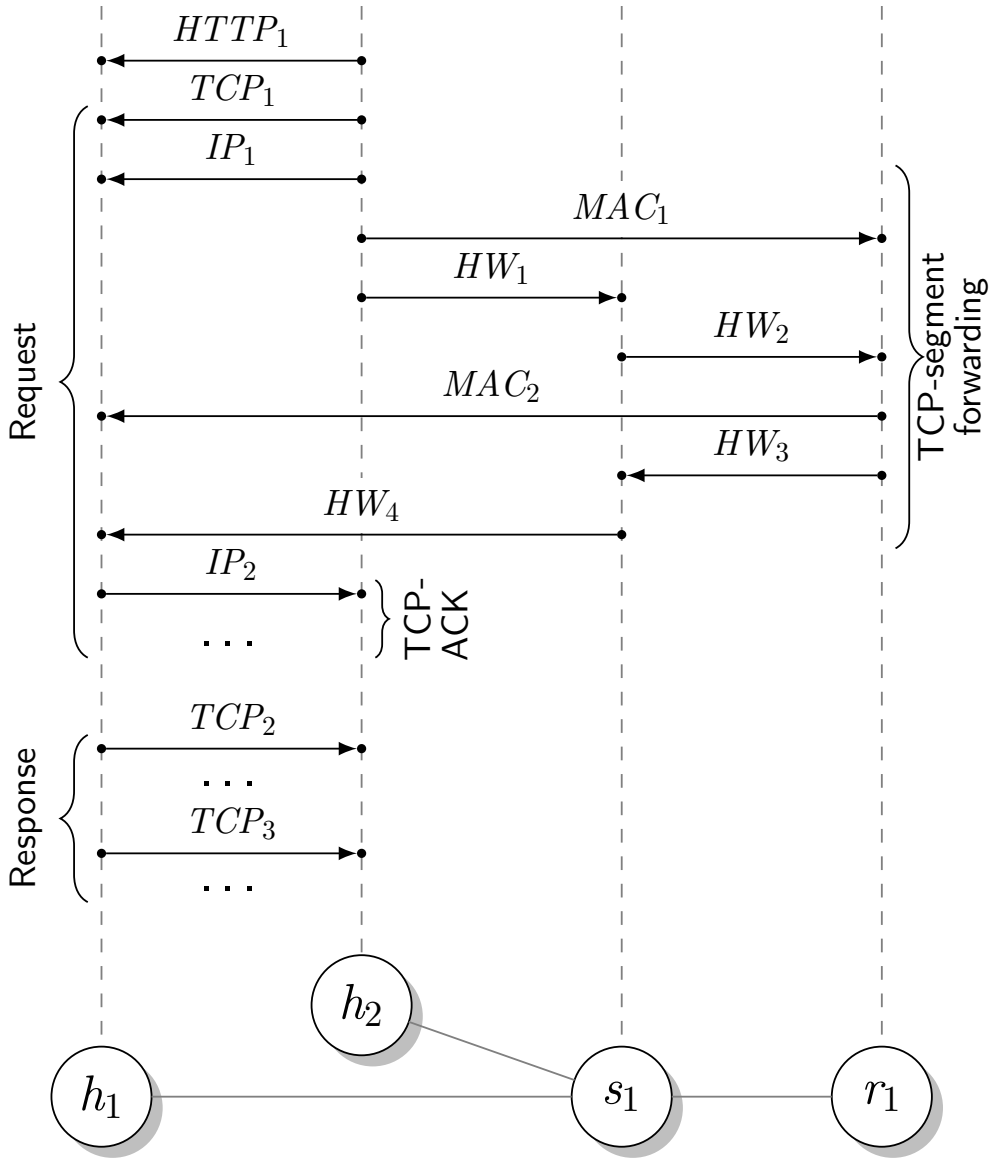


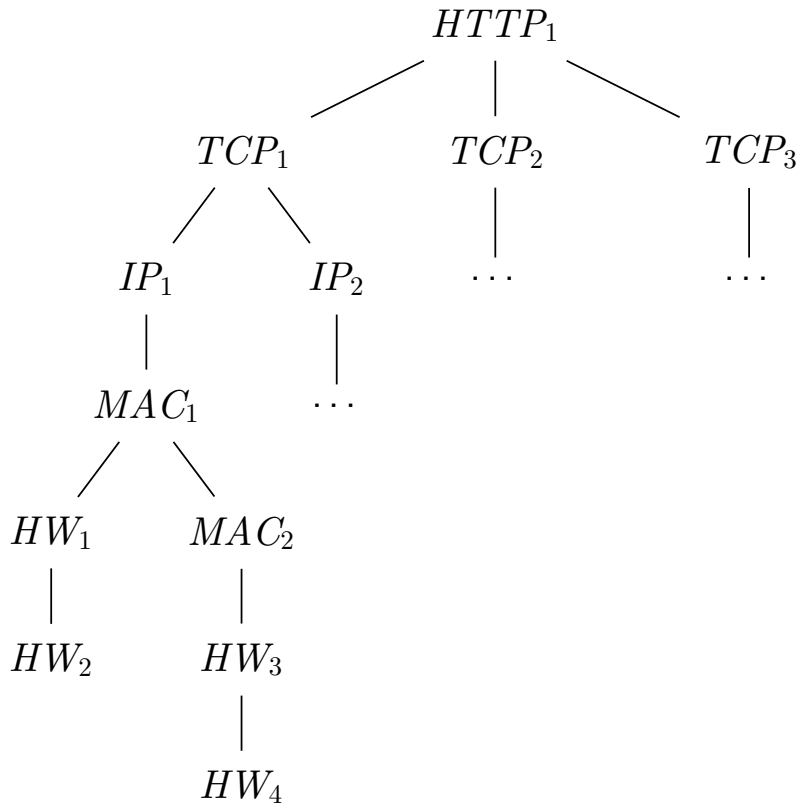
Flows aggregation



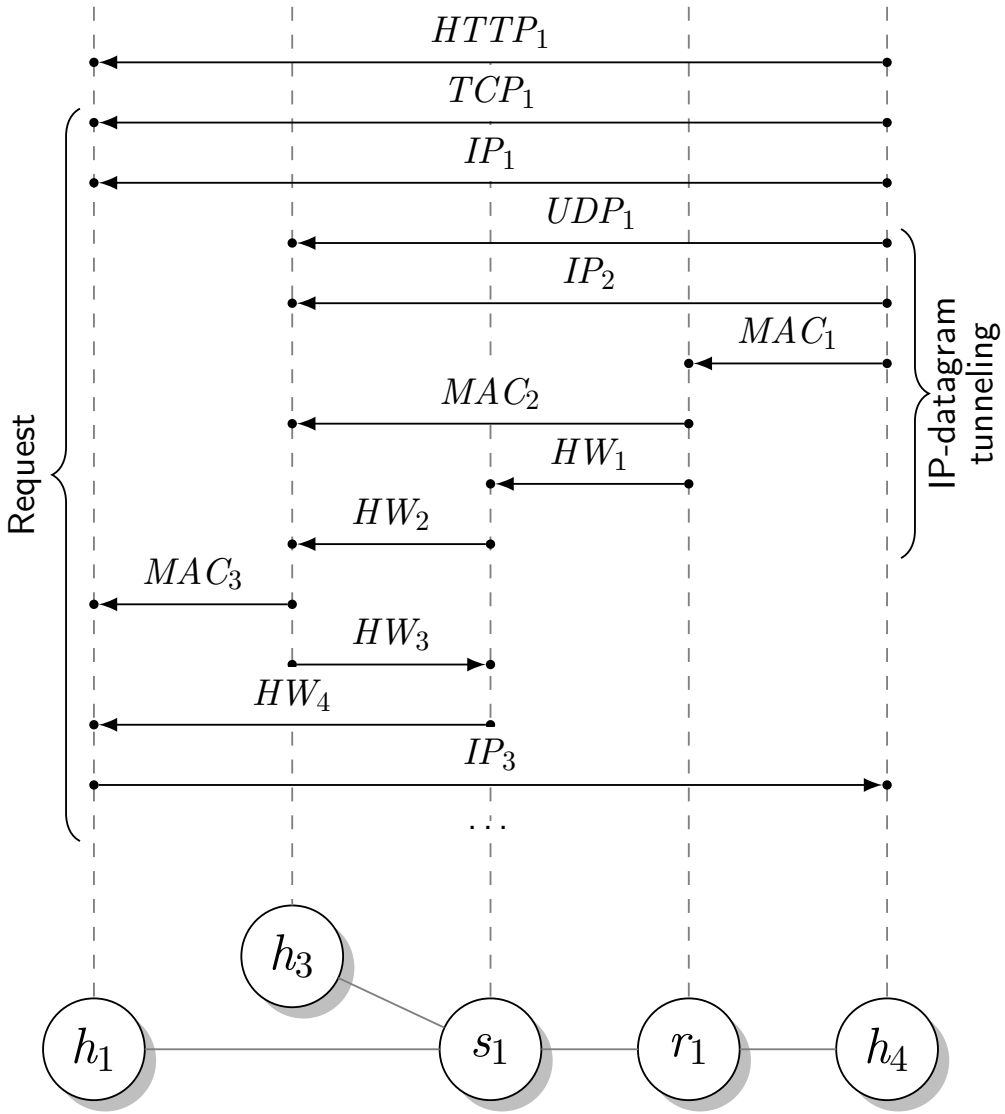


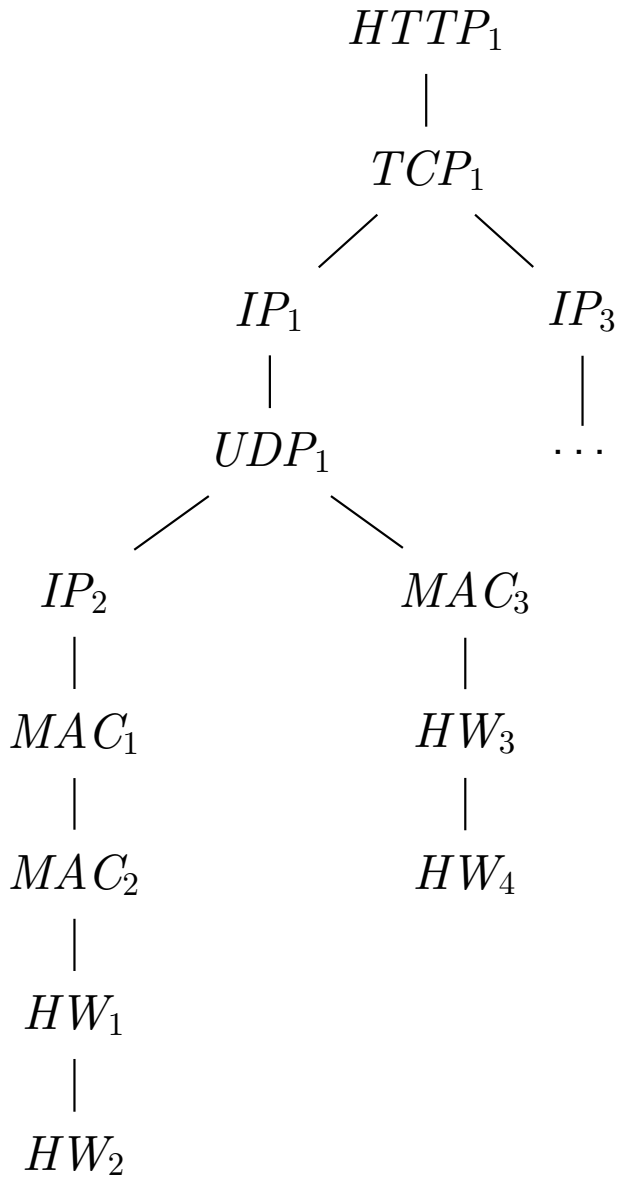
An HTTP-session





An HTTP-session over OpenVPN tunnel





Conclusions

- Modern networks require innovations and deep researches in network management in face of constant complexity growth.
- Enterprise networks have its own specific in network management that is rarely taken into consideration.
- Testbeds are required for new network management approaches evaluation, as well for ad hoc solutions verification in enterprise networks.
- The main challenge is to map traffic data on the enterprise architecture graph.
- A model, describing traffic on basis of incomplete traffic measurements data is proposed:
  - telecommunication process describes any kind of communication between two nodes in the network;
  - tree of telecommunication processes hierarchy could be build;
  - network engineer or software system could query any kind of data flows from this tree.