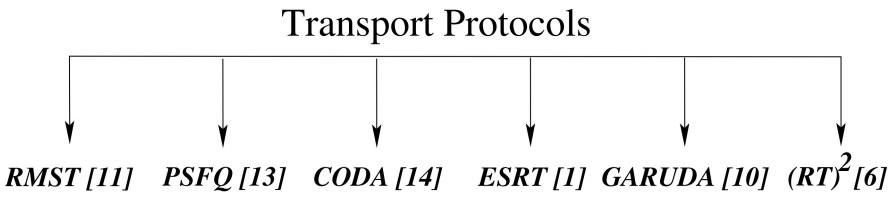


Figure 8.1 Overview of transport protocols.

8.1.4 Biased Implementation

WSNs are usually deployed with a large number of resource-constrained sensor nodes that are connected to a resource-rich sink. The limited processing power and memory capacity of the sensor nodes prevent sophisticated algorithms from being run locally. Hence, the transport layer algorithms should be designed such that most of the functionalities are performed at the sink with minimum functionalities required at the sensor nodes. More specifically, the intelligence should be passed to the sink instead of the sensors. This helps to conserve limited sensor resources and shifts the burden to the high-powered sink.

Moreover, the traffic in the WSNs exhibits significantly different characteristics depending on the flow direction. While the flow in the sensors-to-sink direction may require timely delivery with loss-tolerant operation, the sink-to-sensors direction usually requires a high delivery ratio. Consequently, transport protocols should be designed also by considering these biases in traffic.

8.1.5 Constrained Routing/Addressing

As explained in Chapter 7, wireless sensor nodes may not be assigned unique addresses. Therefore, unlike protocols such as TCP, in the design of transport layer protocols for WSNs the existence of an end-to-end global addressing should not be assumed. It is more likely to have attribute-based naming and data-centric routing, which call for different transport layer approaches.

Several transport layer protocols have been developed for WSNs to address these challenges [15]. An overview of the transport layer solutions that will be discussed next is shown in Figure 8.1. Next, we describe the following protocols: reliable multi-segment transport (RMST), pump slowly, fetch quickly (PSFQ), congestion detection and avoidance (CODA), event-to-sink reliability (ESRT), GARUDA, and real-time and reliable transport $(RT)^2$.

8.2 Reliable Multi-Segment Transport (RMST) Protocol

The RMST protocol is one of the first transport layer protocols developed for WSNs [11]. The main goal of RMST is to provide end-to-end reliability. Accordingly, RMST is built on top of the directed diffusion protocol [7], which is explained in Chapter 7, and uses some of the functionalities of this protocol. More specifically, RMST is designed as a filter that could be attached to the directed diffusion protocol.

RMST provides two of the three functionalities required for a transport layer protocol: reliable transport and multiplexing/demultiplexing. Multiplexing and demultiplexing are carried at the source nodes and the sink, respectively. In the meantime, RMST provides mechanisms to handle errors throughout the routes in the network. To this end, RMST utilizes in-network caching and provides guaranteed delivery of the data packets generated by the event flows.

RMST relies on the directed diffusion routing mechanism for a certain path between a source and destination. Hence, an implicit assumption is that the packets of a flow follow the same path unless there is a node failure. In case of node failures, directed diffusion is assumed to reroute packets. Based on this assumption, RMST has two modes of operation:

- **Non-caching mode:** This mode of operation is very similar to conventional transport layer protocols, where only the source and destination play a role in providing reliability. Consequently, the packet losses are detected at the sink and requested from the source node in an end-to-end fashion through a NACK packet. The advantage of this mode is that it requires no involvement – and, hence, additional processing, storage, and energy consumption – from the intermediate nodes in the multi-hop network.
- **Caching mode:** In this mode, the intermediate nodes on the reinforced path cache the transmitted packets to decrease the overhead in end-to-end retransmissions.

In RMST, each packet of a flow is labeled by a unique sequence number. Accordingly, the packet errors are detected whenever there is a hole in the sequence numbers received. In case of packet errors, nodes request retransmission by sending a NACK packet toward the reverse route from the sink to the sensor, i.e., the *reverse path* as explained in the next examples for the non-caching and caching modes.

EXAMPLE 8.1

Error recovery in non-caching mode is illustrated in Figure 8.2, where a sensor node is trying to transmit a series of packets to the sink through the multi-hop route. The sequence number of the last received packet at the sink is also shown in each case. In non-caching mode, end-to-end retransmissions are performed to provide reliability. In Figure 8.2(a), packet 4 is lost before reaching the sink. The sink can only recognize the packet loss after it receives packet 5 (Figure 8.2(b)). Then, the sink transmits a NACK packet back to the source node as shown in Figure 8.2(c). The lost packet is then retransmitted to the sink through the multi-hop route as shown in Figure 8.2(d).

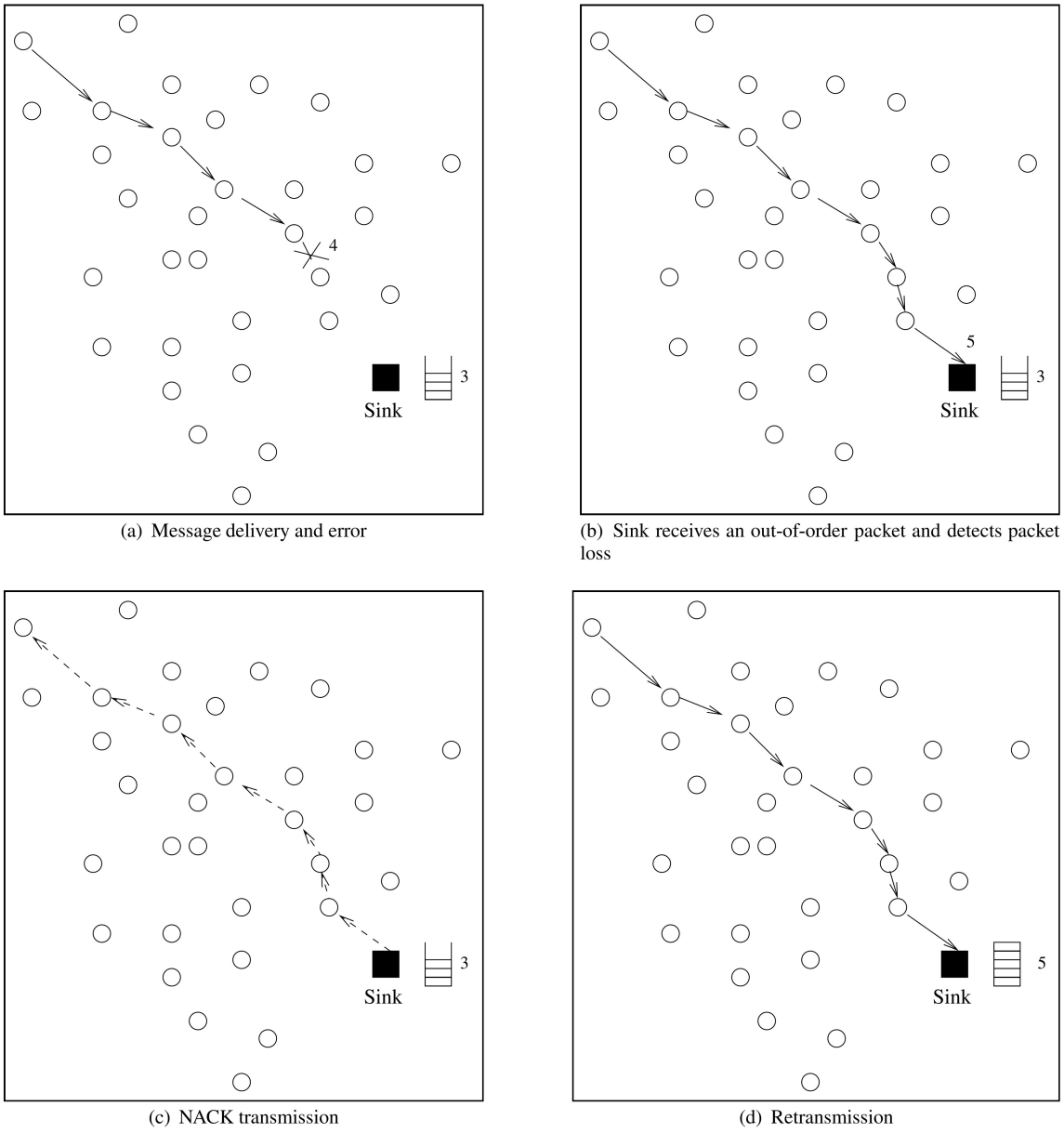
EXAMPLE 8.2

In caching mode, as shown in Figure 8.3, certain sensor nodes are assigned as *caching nodes* (denoted as black nodes in Figure 8.3) on the reinforced path from the sensor node to the sink. In addition to the sink, loss packet detection is also performed at the caching nodes. Similar to the non-caching case, the loss of packet 3 in Figure 8.3(a) can only be detected by the caching node after receiving packet 4 as shown in Figure 8.3(b). Then, the caching node transmits a NACK packet back to the source node as shown in Figure 8.3(c). In this case, as shown in Figure 8.3(d), the first caching node with the missing packet on the reverse path replies and the packets are transmitted to the sink in order. If the packet cannot be found in one of the caching nodes, the NACK packet is propagated until it reaches the source node.

8.2.1 Qualitative Evaluation

In caching mode, RMST essentially creates reliable segments between two consecutive caching nodes and the retransmissions are performed inside these segments instead of through the end-to-end route. As a result, the cost associated with end-to-end retransmissions is minimized. Moreover, RMST aims to provide guaranteed delivery for each flow in the WSN. This is helpful for applications where individual node information is important, such as in network management solutions as discussed in Chapter 9.

RMST may incur additional overhead since caching requires additional processing and memory at the caching nodes. This may increase the overall complexity and energy consumption of the network. Most applications related to event detection/tracking may not require 100% reliability since the individual data flows are correlated and loss tolerant. However, RMST treats each flow separately, which may lead to overutilization of the resources in WSNs. Moreover, guaranteed reliability via in-network caching may bring significant overhead for WSNs with power and processing limitations. Finally, RMST focuses



(a) Message delivery and error

(b) Sink receives an out-of-order packet and detects packet loss

(c) NACK transmission

(d) Retransmission

Figure 8.2 Error recovery with RMST in non-caching mode.

only on the reliability aspects of communication. Since a packet-by-packet reliability notion is followed, a large amount of information may flow inside the network. This will result in congestion and associated packet drops, which is not addressed in RMST.

8.3 Pump Slowly, Fetch Quickly (PSFQ) Protocol

The PSFQ protocol [13] has been developed to address the path from *sink to sensors*. Contrary to many transport layer approaches that focus on the sensors-to-sink path, the reverse path is generally used for network management tasks and retasking of the sensor nodes. Hence, reliability is of major concern.