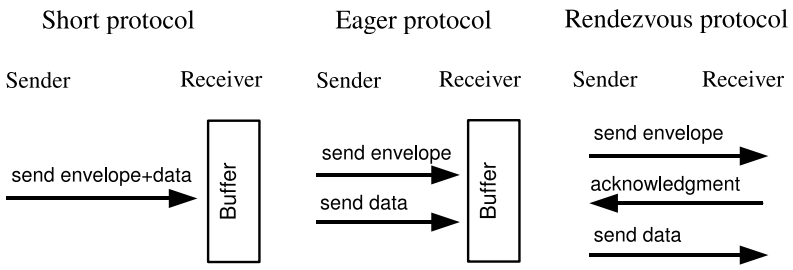

FIGURE 3.3: Three message transfer protocols of MPI

splitting the M subdomain into two equal parts. So in this strategy, we have to accept an additional calculation overhead by splitting the middle subdomain into two parts. If that additional overhead does not balance the advantage of increased overlap, the 5s strategy is preferable.

3.3 Message transfer protocols of MPI

MPI defines three protocols to transfer messages with different message sizes: short, eager, and rendezvous [50, 76]. The short and eager protocols are used to transfer short messages, and the rendezvous protocol is used for long messages. The aim of using different methods is to balance the latency and the bandwidth. For a small message the bandwidth is less important than the latency. If a message is long, the bandwidth is more important.

Figure 3.3 describes the procedure of these protocols. An MPI message consists of two parts: data to be sent/received and an envelope that contains information needed to route data. In the eager protocol, both the envelope and data are sent immediately to the destination and the sending buffer is immediately returned to the application. Since the message is sent without knowing if there is a matching receive waiting, the message may need to be stored in a buffer at the destination, till a matching receive is posted by the application. The transferring approach using the eager protocol is called non-blocking MPI. The short protocol is a special variation of the eager protocol where the envelope and data all fit in one packet, the smallest unit of data that is passed through the network. The maximum amount of data that can be sent in one packet is 120 bytes [86].

In the rendezvous protocol, when a send is posted, the envelope is sent to the receiver and eventually buffered there. When the receive is posted and the appropriate envelope has already been received, the receiver sends an acknowledgment to the sender. Data is sent only when the sender received this confirmation. This protocol requires the receiver to eventually buffer only the

TABLE 3.2: DAS-3 resources

Cluster	#Nodes	Type	Speed	Memory	Storage
VU	85 dual	Dual core	2.4 GHz	4 GB	10 TB
LU	32 dual	Single core	2.6 GHz	4 GB	10 TB
TUD	68 dual	Single core	2.4 GHz	4 GB	5 TB
UvA-VLe	41 dual	Dual core	2.2 GHz	4 GB	5 TB
UvA-MN	46 dual	Single core	2.4 GHz	4 GB	3 TB

envelopes. Because the receiver is synchronized, the rendezvous protocol can avoid an additional copy of the data. This transferring method is called blocking MPI.

As opposed to blocking MPI, where the sender is blocked until the receiver comfirms that the message has been completely received, a send by non-blocking MPI returns control immediately to the application. Therefore, other tasks such as calculations can be done while data is transferring. Because of buffer requirement, the use of non-blocking MPI is limited by the fact that the message has to be smaller than the buffer size, which has a maximum of 64 KB [50]. MPI switches to blocking mode automatically if the message size is larger than 64 KB.

3.4 Experiment configuration

3.4.1 Hardware

The experiments were conducted on DAS-3 [67]. DAS-3 is a wide-area distributed system which consists of five clusters located at four universities in The Netherlands: Vrije Universiteit (VU), Leiden University (LU), University of Amsterdam (UvA-VLe and UvA-MN), and Delft University of Technology (TUD). The computational resources of DAS-3 are presented in Table 3.2.

Single cluster experiments were performed on the VU cluster, which has 85 dual compute nodes, so 170 processors in total, each dual core. The clock frequency of each processor is 2.4 GHz. Our experiments on two clusters involved the VU cluster and the LU cluster. The latter has 32 dual nodes single core processors, with a clock frequency of 2.6 GHz. In all experiments we always use both clusters in single node, single core mode. The physical distance between the two clusters is approximately 40 km. The two-cluster configuration is considered as a proto-type for a grid [22].