

CHAPTER 2 RELATED WORK

Proxy caching is one of the techniques to improve the throughput performance in a WMN [3], [4]. In mesh proxy caching, each mesh router caches files or a part of files similar to a web proxy but the caches are distributed (among mesh routers) throughout the network rather than at one specific node. This technique is adapted in the MeshCache project [3], referred to as hop-by-hop caching or on-path caching.

Proxy caching in a WMN can be achieved by enabling the caching to function at the mesh routers. For example, in the MeshCache project [3], all mesh routers cache data passing through them, called on-path caching. A mesh router may also cache data overhearing in the air, called overhearing caching [4].

Let us consider Figure 2.1 as an example of on-path caching. Client A is attached to mesh router R1 and needs to download a file, which could be some recent OS update file, from the Internet. Its file request is sent to R1. If R1 has no cache, it will forward the request on behalf of the client to its upstream mesh router R2 on the route to the gateway according to the routing table. The request is still forwarding if a cache is not hit in the intermediate mesh router until reaching the gateway, in which case the file is retrieved directly from the Internet. The file is returned to client A and all intermediate mesh routers cache the file. Later, if one of these mesh routers receives a request from this same file, the cache hit will occur and the requested file will be returned immediately. Note that mesh routers R4 and R5 are not the data path and hence do not cache anything. On-path caching exploits significant traffic passing through them to improve data throughput but it has to consume more power on each network element.

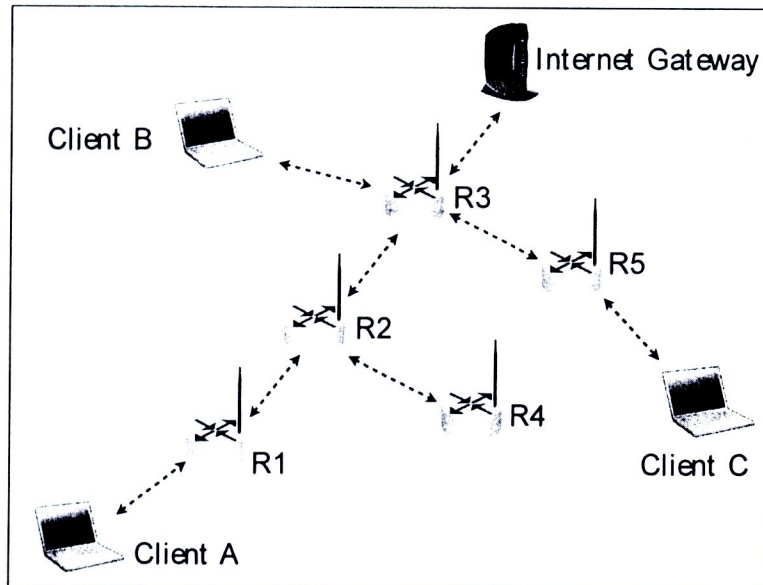


Figure 2.1 Example of a Wireless mesh network. R1, R2, R3, R4, R5 are mesh routers, with the dash lines indicating that nodes are within communication range of each other. Client A is attached to R1, client B to R3, and client C to R5 respectively.

Overhearing caching adapted by Ditto system [4] extends on-path caching with opportunistic listening to improve the throughput. From the previous example of on-path caching discussed earlier, let us assume in Fig. 1 that mesh router R4 and R5 are in the communication range of R2 and R3 respectively. When a requested file is transmitted by R3, R5 can overhear the file and take a chance to cache it. It is also the same for R4 when R2 transmits a file. Consequently, if client C that attached to R5 requests the same file as client A, the cache hit will occur at R5 and the file will be returned to client C. However, mesh nodes in Ditto cache the chunks of a file rather than a whole complete file, and use content-based hashing to identify the chunks to transfer as in work in [4].