|  |  |  |
| --- | --- | --- |
| 函数名 | 返回值 | 功能 |
| \_\_kaodv\_queue\_enqueue\_entry(struct kaodv\_queue\_entry \*entry) | int | 静态内联函数，使当前表进入排队队列 |
| \_\_kaodv\_queue\_find\_entry(kaodv\_queue\_cmpfn cmpfn, unsigned long data) | kaodv\_queue\_entry\* | 查找并返回与 cmpfn 匹配的排队条目，或者如果 cmpfn 为 NULL，则返回最后一个条目 |
| \_\_kaodv\_queue\_find\_dequeue\_entry(kaodv\_queue\_cmpfn cmpfn, unsigned long data) | kaodv\_queue\_entry\* | 查找并把与cmpfn匹配的路由表从排队队列中去除并返回该路由表 |
| \_\_kaodv\_queue\_flush(void) | void | 把排队队列清空 |
| \_\_kaodv\_queue\_reset(void) | void | 重置函数，调用flush函数 |
| kaodv\_queue\_find\_dequeue\_entry(kaodv\_queue\_cmpfn cmpfn, unsigned long data) | kaodv\_queue\_entry  \* | 写保护，找到删除的表 |
| kaodv\_queue\_flush(void) | void | 写保护，调用flush函数清空队列 |
| kaodv\_queue\_enqueue\_packet(struct sk\_buff \*skb, int (\*okfn) (struct sk\_buff \*)) | int | 写保护，把当前信息的表写入队列 |
| dest\_cmp(struct kaodv\_queue\_entry \*e, unsigned long daddr) | int | 判断参数中的表是否和参数中的路由地址匹配 |
| kaodv\_queue\_find(\_\_u32 daddr) | int | 读保护，根据地址找到表 |
| kaodv\_queue\_set\_verdict(int verdict, \_\_u32 daddr) | int | 根据verdict来采用不同处理，如果是删除表则发送ICMP消息告知目的节点已不可达，如果是发送消息则根据地址查找表，找得到并且该节点是网关则对IP报文进行封装后重新查找路由，找不到就释放空间 |
| kaodv\_queue\_get\_info(char \*buffer, char \*\*start, off\_t offset, int length) | int | 版本低于2.6.24的处理函数，得到队列长度信息 |
| kaodv\_queue\_get\_info(char \*page, char \*\*start, off\_t off, int count,int \*eof, void \*data) | int | 版本高于2.6.24的处理函数，得到队列长度信息 |
| init\_or\_cleanup(int init) | int | Proc文件系统用于访问内核，该函数先进行队列、proc文件初始化，再创建新的proc函数并进行配置 |
| kaodv\_queue\_init(void) | int | 如果已经初始化调用该函数进行proc文件初始化 |
| kaodv\_queue\_fini(void) | void | 如果没有初始化调用该函数进行proc文件初始化 |